UNIVERSIDADE FEDERAL DE SANTA CATARINA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE AUTOMAÇÃO E SISTEMAS

Maiquel de Brito

A MODEL OF INSTITUTIONAL REALITY SUPPORTING THE REGULATION IN ARTIFICIAL INSTITUTIONS

Florianópolis

2016

Maiquel de Brito

A MODEL OF INSTITUTIONAL REALITY SUPPORTING THE REGULATION IN ARTIFICIAL INSTITUTIONS

Tese submetida ao Programa de Pós-Graduação em Engenharia de Automação e Sistemas da Universidade Federal de Santa Catarina para a obtenção do Grau de Doutor.
Prof. Dr. Jomi Fred Hübner: Orientador
Universidade Federal de Santa Catarina
Prof. Dr. Olivier Boissier: Coorientador
École Nationale Supérieure des Mines de Saint-Etienne

Florianópolis

2016

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Brito, Maiquel de
A model of institutional reality supporting the regulation in artificial institutions / Maiquel de Brito ; orientador, Jomi Fred Hübner ; coorientador, Olivier Boissier. - Florianópolis, SC, 2016.
175 p.

Tese (doutorado) - Universidade Federal de Santa Catarina, . Programa de Pós-Graduação em Engenharia de Automação e Sistemas.

Inclui referências

1. Engenharia de Automação e Sistemas. 2. Sistemas multiagente. 3. Instituições. 4. Normas. I. Hübner, Jomi Fred. II. Boissier, Olivier. III. Universidade Federal de Santa Catarina. Programa de Pós-Graduação em Engenharia de Automação e Sistemas. IV. Título.

Maiquel de Brito

A MODEL OF INSTITUTIONAL REALITY SUPPORTING THE REGULATION IN ARTIFICIAL INSTITUTIONS

Esta Tese foi julgada aprovada para a obtenção do Título de “Doutor”, e aprovada em sua forma final pelo Programa de Pós-Graduação em Engenharia de Automação e Sistemas da Universidade Federal de Santa Catarina.

Florianópolis, 12 de setembro 2016.

Prof. Dr. Daniel Coutinho
Coordenador
Universidade Federal de Santa Catarina

Orientador
Universidade Federal de Santa Catarina
Prof. Dr. Jomi Fred Hübner

Coorientador
École Nationale Supérieure des Mines de Saint-Etienne
Prof. Dr. Olivier Boissier

Banca Examinadora:

Prof. Dr. Antônio Carlos da Rocha Costa
Universidade Federal do Rio Grande

Prof. Dr. Gustavo Alberto Giménez Lugo
Universidade Tecnológica Federal do Paraná

Profa. Dra. Jerusa Marchi
Universidade Federal de Santa Catarina

Prof. Dr. José Eduardo Ribeiro Cury
Universidade Federal de Santa Catarina

Prof. Dr. Marcelo Ricardo Stemmer
Universidade Federal de Santa Catarina

à Cinthia,
parte indissolúvel
de tudo o que realizei.

AGRADECIMENTOS

Antes de mais nada, agradeço ao meu orientador, Professor Jomi F. Hübner, por ter aceitado me acompanhar neste trabalho ao longo de quatro anos de doutorado. Agradeço por sua confiança em meu trabalho, por sua solicitude ao prestar todo tipo de auxílio, pelo valor prestado às ideias relevantes, pelo respeito ao alertar sobre os equívocos e pela disposição nas inúmeras revisões do texto da tese e dos artigos que foram escritos. Para além de toda a colaboração na elaboração desta tese, agradeço por todo o aprendizado em questões relacionadas à pesquisa científica de forma geral, bem como pelo incentivo e oportunização de colaboração com uma comunidade acadêmica mais ampla.

Sou também grato ao Professor Olivier Boissier, que se tornou coorientador desta tese, pela colaboração ao longo de praticamente todo o período de doutorado. Agradeço-o, bem como à sua família, por terem recebido a mim e à minha esposa em sua casa, dando todo o suporte para nossa instalação, adaptação e permanência em Saint-Etienne (França). Esse auxílio foi fundamental para a realização de um estágio de um ano na École Nationale Supérieure des Mines de Saint-Etienne. O Professor Olivier oportunizou, entre outras coisas, a cooperação com a colega Lauren Thévin e com a Professora Catherine Garbay, da Universidade de Grenoble. A elas agradeço pela valiosa colaboração nos trabalhos relacionados ao cenário de gerenciamento de crises, que foi importante de várias maneiras ao longo do desenvolvimento deste trabalho.

Agradeço às institutições que forneceram suporte financeiro para a realização deste trabalho: à CAPES, pela bolsa de doutorado e pela bolsa PDSE (4926-14-5), que viabilizou meu estágio na França, bem como e ao Programa de Pós-Graduação em Engenharia de Automação e Sistemas (PPGEAS) e à Pró-Reitoria de Pós-Graduação (PROPG) da UFSC que, sempre que necessário, viabilizaram minha participação em eventos tanto no Brasil quanto no exterior. Agradeço à École Nationale Supérieure des Mines de Saint-Etienne por ter me recebido e durante um ano, fornecendo todos os recursos necessários para o desenvolvimento de meu trabalho e, quando necessário, juntamente com outros órgãos, tais como ARMINES e Communautés de Recherche Académique Rhône-Alpes, viabilizado minha participação em eventos na Europa.

Este trabalho contou com o o auxílio da comunidade envolvida no desenvolvimento de ferramentas relacionadas à programação de sistemas multiagente. Agradeço em especial àqueles envolvidos no desenvolvimento do *framework* JaCaMo pelas colaborações. Agradeço também ao Professor Alessandro Ricci (Universidade de Bologna - Itália), pelo suporte prestado na utilização e adaptações do CArtAgO, que foram essenciais para as implementações realizadas ao longo deste trabalho.

Agradeço aos amigos Tiago, Daniela, Maicon e Fernando, companheiros de grupo de pesquisa na UFSC, pela parceria em muitos e diferentes momentos, incluindo tanto aqueles em que discutimos ideias contidas nessa tese quanto aqueles em que essas discussões foram trocadas pela leveza de conversas descontraídas.

Agradeço à minha família por todo apoio, compreensão e confiança. À minha mãe, Marcia (*in memorian*), que não conseguiu ver o final deste trabalho. E, principalmente, agradeço à minha esposa, Cinthia, que encarou esse projeto como se fosse seu, valorizando as conquistas que nem eu mesmo percebia e compreendendo as muitas e necessárias ausências.

Por fim, agradeço a Deus, que me concedeu cada um dos dias necessários à realização desse trabalho.

RESUMO EXPANDIDO

UM MODELO DE REALIDADE INSTITUCIONAL PARA SUPORTAR A REGULAÇÃO EM INSTITUIÇÕES ARTIFICIAIS

Introdução. Sistemas multiagente abertos requerem algum tipo de regulação para conciliar a atuação autônoma dos agentes com os objetivos globais do sistema. Normas são uma forma usual de expressar os requisitos de regulação nesse tipo de sistema e são normalmente especificadas em termos abstratos que podem ser vistos como uma interpretação dos elementos concretos que compõem o sistema, os quais fazem parte do *ambiente* em um sistema multiagente. Por exemplo, em uma norma especificando que clientes são obrigados a pagar por suas compras, agentes podem ser considerados *clientes* e depósitos bancários podem ser considerados *pagamentos*. Esses conceitos abstratos fazem parte da *realidade institucional* das sociedades de agentes. Essa realidade não é composta, mas *constituída* a partir dos elementos existentes no ambiente. Por exemplo, um agente atuando no ambiente pode constitutir um *cliente* na realidade institucional. Para que a regulação especificada pelas normas possa ser baseada na realidade institucional, é necessário haver meios para especificá-la, definindo sua constitutição a partir do ambiente.

Objetivos. O objetivo desta tese é propor um modelo de realidade institucional, constitutída a partir do ambiente, para suportar a regulação especificada pelas normas em sistemas multiagente. Para atingir esse objetivo, toma-se, como inspiração, a teoria do filósofo John Searle, que observa que, nas sociedades humanas, a realidade institucional existe devido à concordância coletiva de que alguns elementos existentes no mundo contam como elementos da realidade institucional. Por exemplo, concorda-se que cédulas de papel com determinadas características contam como dinheiro. Para atingir o objetivo desta tese, é necessário definir quais são as abstrações utilizadas para representar a realidade institucional em instituições artificiais, bem como definir como a heterogeneidade dos elementos do ambiente é capturada pela realidade institucional, definindo ainda como as normas, que podem ser concebidas de diferentes maneiras, podem ser acopladas em uma representação unificada de realidade institucional.

Contribuições. Esta tese propõe um modelo de realidade institucional para suportar a regulação em sociedades de agentes, definindo suas representações bem como sua dinâmica, que é consequência da dinâmica do ambiente. Essa realidade institucional é representada através de *status functions*, que são funções atribuídas pela instituição a agentes atuando, eventos ocorrendo e estados vigorando no ambiente. A partir desse modelo, define-se uma linguagem para especificar a realidade institucional. Além disso, define-se como as normas, da forma em que são propostas na literatura, baseiam sua regulação nessa realidade institucional ao fazerem referência às *status functions*. Esses elementos permitem conceber um modelo de instituição artificial, chamado SAI (de *Situated Artificial Institution* ou *Institução Artificial Situada*), em que é possível especificar a realidade institucional para suportar a regulação provida por normas que seguem diferentes modelos normativos. A partir desse modelo, propõe-se a arquitetura de uma plataforma para execução dessas instituições em sistemas multiagente. O modelo proposto é avaliado através de exemplos de aplicações e, quando possível, comparado com outras propostas de modelos institucionais.

Conclusões. O modelo de realidade institucional proposto concebe abstrações apropriadas para a representação desse aspecto social dos sistemas multiagente. A partir desse modelo, é possível conceber instituições artificiais em que a regulação é baseada em uma realidade institucional constituída a partir do ambiente. Tem-se, assim, instituições em que regulação e realidade institucional são desacopladas porém coerentemente conectáveis.

Palavras-chave: Sistemas multiagente. Instituições. Normas.

ABSTRACT

Open Multi-Agent Systems require some regulation to conciliate the autonomy of the agents with the overall goals of the system. Norms are a usual way to express the regulative requirements in this kind of system and are usually specified in abstract terms that can be seen as an interpretation of the concrete elements composing the system. For example, in the norm "*buyers are obliged to pay for their purchases*", agents can be seen as *buyers* and bank transfers can be seen as *payments*. These abstract concepts, such as *buyers* and *payments*, refer to elements belonging to the *institutional reality* in a society of agents. Such reality is not composed, but *constituted* from the elements existing in the environment where the agents act. This thesis proposes a model of institutional reality to base the regulation of agent societies, defining its representations as well as its dynamics, that is constituted from the dynamics of the environment. From this model, it is defined a language to design the institutional reality. Furthermore, it is defined how the norms, as conceived by the literature, can base their regulation on this institutional reality. These elements enable us to conceive a model of artificial institution called SAI (from Situated Artificial Institution), that considers both the institutional reality and norms, that can follow different normative models. From this model, it is proposed a platform architecture to deploy these institutions in Multi-Agent Systems. The proposed model is evaluated through application examples and, when possible, compared with other proposals of institutional models.

Keywords: Multi-Agent Systems, Institutions, Norms.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

AI	Artificial Intelligence
MAS	Multi-Agent Systems
PhD	Doctor of Philosophy
ADICO	Attribute, Deontic, Aim, Conditions, and Or-else
OOP	Object Oriented Programming
SAI	Situated Artificial Institutions
SFA	Status Function Assignment
GIS	Geographic Information Systems
RFID	Radio-Frequency Identification
BNF	Backus-Naur Form

LISTA DE SÍMBOLOS

$\mathcal{X}$	Set of environmental elements in an institutional specification
$\mathcal{A}_{\mathcal{X}}$	Set of agents in an institutional specification
$\mathcal{E}_{\mathcal{X}}$	Set of events in an institutional specification
$\mathcal{S}_{\mathcal{X}}$	Set of environmental properties in an institutional specification
$\mathcal{F}$	Set of status functions of an institution
$\mathcal{A}_{\mathcal{F}}$	Set of agent-status functions of an institution
$\mathcal{E}_{\mathcal{F}}$	Set of event-status functions of an institution
$\mathcal{S}_{\mathcal{F}}$	Set of state-status functions of an institution
$\mathcal{C}$	Set of constitutive rules of an institution
λ	Omitted element in a constitutive rule
W	Set of w-formulae
X	Actual environmental state
A_X	Set of agents participating in the system
E_X	Set of events occurring in the environment
S_X	Set of environmental properties
A_F	Set of agent-status function assignments of an institution
E_F	Set of event-status function assignments of an institution
S_F	Set of state-status function assignments of an institution
F	Constitutive state of the institution
N	Normative state
SAI_{Dyn}	SAI state
$\mathbb{N}$	Set of the natural numbers
SAI^{i}_{Dyn}	SAI state at the i^{th} step of its history
θ	Substitution of variables
F^{*}	Closure of status function assignments
$W_{\mathcal{F}}$	Set of sf-formulae
W_{N}	Set of n-formulae

SUMÁRIO

1 INTRODUCTION

Technological advances in the recent decades, such as the advent of Internet, made computation to move from just data storage and processing by single, standalone units, to what some authors name *computation as interaction*, where computational entities interact among themselves and with humans (LUCK et al., 2005). For example, in 2009, 17 of the top 20 most prolific *Wikipedia* editors were autonomous programs named *bots*, that performed 16,33% of all the edits (GEIGER, 2009, 2011). As another example, in 2015, around 70% of the total trading volume in major European and US equity exchanges involved algorithmic traders, part of them autonomous (FARJAM; KIRCHKAMP, 2015).

Artificial Intelligence (AI) in general and Multi-Agent Systems (MAS) in particular have provided means to design autonomous, goal-oriented artificial entities – the *agents* – to participate in this kind of open, decentralized system. In this context, *open* MAS are systems where agents can freely enter and exit (BOISSIER; HÜBNER; SICHMAN, 2007; DEMAZEAU; COSTA, 1996), can be designed and implemented by different parties (FORNARA; VIGANÒ; COLOMBETTI, 2004), and it is not possible to predict, in design time, neither the number, nor the behaviour, nor the way the agents shall interact among themselves and explore the available resources (PIUNTI, 2009).

Open MAS can be seen as *agent societies* composed of heterogeneous, autonomous, self interested individuals with possibly conflicting personal interests, limited trust and unexpected behaviours (ARTIKIS; PITT; SERGOT, 2002). For these reasons, it is necessary to regulate the agents' behaviour to conciliate their autonomy with the overall systems' expectations (MOSES; TENNENHOLTZ, 1995; CASTELFRANCHI, 2000; FASLI, 2004).[1] Designing open MAS, thus, involves not only to design the agents involved in the interactions. Rather, in order to provide some regulation to the system, it involves also the design of the social requirements to be followed by the participating agents (DASTANI; DIGNUM; DIGNUM, 2003). These social requirements are usually expressed through *norms* (LÓPEZ; LUCK; D'INVERNO, 2002; CRIADO; ARGENTE; BOTTI, 2011; FORNARA et al., 2008).

Independent of the normative regulation, the arena of actions

---

[1] Regulation, in this thesis, consists not only of to constrain the actions of the agents. More than that, regulation also aims to help the agents to properly cooperate even in systems to which they have not been specifically designed.

of the agents is, by definition, the *environment* (RUSSELL; NORVIG, 2003; MAES, 1995; WOOLDRIDGE; JENNINGS, 1995; FRANKLIN; GRAESSER, 1997). The environment is usually conceived as the set of non-autonomous elements that are perceived and acted upon by the agents, where they act to achieve their goals (e.g. sensors and actuators, printers, networks, databases, web services, etc.) (RUSSELL; NORVIG, 2003; WEYNS; OMICINI; ODELL, 2007; RICCI; PIUNTI; VIROLI, 2011). From a normative perspective, the agents are also part of the environment.

Environments are typically dynamic, changing along the execution of the system (HELLEBOOGH et al., 2006). The state of the existing elements may change, new elements may be added, etc. Designing regulation in such a dynamic environment tends to be complex. For example, in an auction scenario where the agents are expected to bid, the norm "*bob is obliged to raise his hands*" does not cover neither obligations for the agent *tom* nor bids by utterances. The norm should be changed every time a new agent is targeted by the obligation and every time a new bid method is added (e.g. "*bob and tom are obliged to either raise their hands or utter their offer*"). For this reason, the regulation is usually specified referring to abstract concepts more related to the domain of the application – such as "*bidder*" and "*bid*" – instead of referring to the concrete elements involved in the interactions, that can be even unknown when the regulation is specified (ALDEWERELD et al., 2010; GROSSI; DIGNUM, 2005). For example, a norm stating that "*bidders are obliged to bid*" can range over all the agents that could be considered as *bidder* and over all the actions considered as *bid*, independent of their concrete realisation in the environment. This norm does not require changes when the set of agents considered as *bidder* and the set of possible bidding methods change.

The elements referred by the norms, such as *bid*, *bidder*, etc. are part of the *institutional reality* of agent societies. The institutional reality, that can be seen as an interpretation of the environment under regulation, is constituted from the elements placed there (BOELLA; TORRE, 2004; MALUCELLI; CARDOSO; OLIVEIRA, 2005; FORNARA; VIGANÒ; COLOMBETTI, 2007). For example, agents acting in the environment may constitute (or *count as*) *bidders* in the institutional reality while actions of these agents, performed in the environment, may count as *bids*. The existence of an institutional reality to which norms refer brings to MAS the notion of *artificial institution* (or simply *institution*). In spite of the different existing definitions of *institution*, this work considers it as an element that has the role of housing the norms governing the agents' activities but has also the prior role of enabling the existence of the institutional reality that supports the norms (SEARLE, 1995, 2009; RUITER, 1997; JONES; SERGOT, 1993; MALUCELLI; CAR-

DOSO; OLIVEIRA, 2005; BOELLA; TORRE, 2004; BROERSEN et al., 2013; BROERSEN; TORRE, 2012; BALKE et al., 2013). That is to say, from a system composed of entities such as agents, web services, messages, databases, etc. (ultimately conceived to be deployed as software pieces), the institution brings into the system elements such as bidders, bids, auctions, etc. supporting norms such as "*bidders are obliged to bid*".

1.1 PROBLEMS

Institutional reality is a social phenomenon observed in human societies. It is the portion of reality composed of abstract elements such as bids, money, presidents, etc. that arise from a concrete world composed of utterances, pieces of paper, people, etc. The behaviour of the individuals in the societies is based on this institutional reality (SEARLE, 1995, 2009; HART et al., 2012). Individuals buy goods because some paper bill counts as *money* and follow some person because he counts as *president*. The value is in the notion of *money* instead of being in the paper bill, the leadership is in the position of *president* instead of being the in the individual itself, and so on.

While institutional reality is a phenomenon inherent to human societies, it requires some modelling in artificial ones. Having explicit representations of the institutional reality arising from the environment to support the normative regulation is a key issue to design and manage regulation of agent societies (BROERSEN et al., 2013; BROERSEN; TORRE, 2012; BOELLA; TORRE, 2004). Rephrasing the quote "from protons to presidents" (SEARLE, 2009, page 3), we can say that agent societies need to go "from bits to bids" and then to obligations, prohibitions, etc. The problem addressed in this thesis is precisely the *design of artificial institutions that not only specify the regulation but that also have means to specify the institutional reality which the regulation is based on, arising from the concrete environment where the agents act*.

From the outlined, we can see it as a threefold problem (cf. Figure 1). It requires to define (i) what are the appropriate abstractions to represent the institutional reality, (ii) how the regulation represented by the norms is coupled with such representations and (iii) how the complex, heterogeneous, highly dynamic environmental elements give rise to elements within the institutional reality.

Regarding to the first point, although the institutional reality is somehow related to the regulation of MAS, it cannot be represented through *norms* as it is not supposed to express the expected agents' behaviour. In other hand, although it arises from the environment, it is not supposed to be the media to

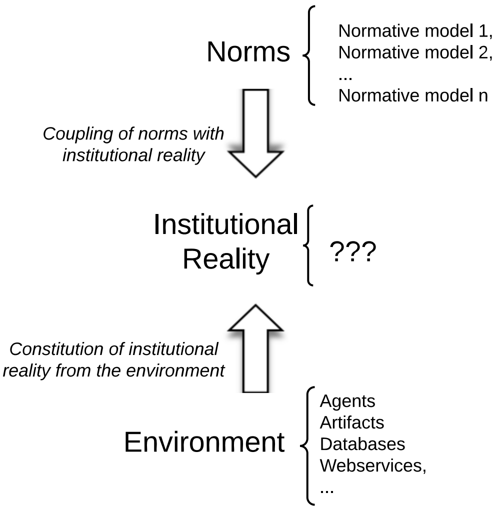


Figura 1 – Decomposition of the problem

support the agents actions and, thus, it is not properly expressed through environmental abstractions. It cannot, naturally, be expressed through the agent metaphor because it is not composed of autonomous elements having beliefs, intentions, goals, etc. Some current proposals conceive the institutional reality as the vocabulary to be used by the norms, containing words such as *bid*, *bidder*, *auction*, etc. But these approaches do not conceive the abstractions that are the building blocks of the institutional reality. Other approaches consider that the institutional reality is composed of facts related to the dynamics of the norms, such as *activation*, *violation*, *fulfilment*, etc. But in this case the representation of the institutional reality is mixed with the representation of the norms.

Regarding to the second point, it is necessary to consider that norms can express the expected agents' behaviour in many different ways. For example, some normative models express norms only as obligations (HÜBNER; BOISSIER; BORDINI, 2011) while others consider also prohibitions (TINNEMEIER et al., 2009) and permissions (OREN et al., 2009); or, for instance, some models consider that the agents can repair their misbehaviour (PANAGIOTIDI; ÁLVAREZ-NAPAGAO; VÁZQUEZ-SALCEDA, 2013) while others consider that they are sanctioned (TINNEMEIER et al., 2009). Norms following different models can model different regulative requirements even in the same system (CRIADO; ARGENTE; BOTTI, 2011). Despite this diversity, all the norms following the different models in a system should be based on the same institutional reality. For example, every norm referring to *bidder* and *bid* should share the same notion about these concepts, including their constitution from the environment. The representation of the institutional reality thus cannot be mixed with the representation of the norms, otherwise there could have as many representations of the institutional reality as many are the normative models involved in the regulation of the system.

A third point to consider is the fact that the institutional reality does not exist by itself. It arises from the environment where the agents act, that is composed of elements of different natures and dynamics. Enabling the institutional reality requires to take into account these natures and dynamics. For example, *bidder* must be the institutional counterpart of agents acting in the environment while *bid* must be the institutional counterpart of fact occurring there. This relation between the semantics of the elements of the institutional reality and their counterpart in the environment must be also explicit. Otherwise, one could specify that *bid* corresponds to agents in the environment while *bidder* corresponds to their actions. This relation must be also observed in the management of the institutional reality. Elements corresponding to agents (e.g. *bidder*) should exist in the institutional reality only while the agents are participating in the system. Elements corresponding to facts (e.g. *bid*) should exist only while the corresponding fact is occurring in the environment. This applies to all the different kinds of elements possibly composing the institutional reality.

As we can see, the notion of institutional reality is present, implicitly at least, in the regulation of MAS, that is mainly represented through norms. Normative design and monitoring is a well addressed topic in MAS research, with many proposals of formalisms, models, languages, tools, etc. However, the institutional reality (with some abuse of this expression) is addressed through simplistic solutions such as *ad hoc* links between regulative representations and the environment or a vocabulary without semantics within the institution. Some problems of these approaches are their support to single specific models of norms, the impossibility of to properly link concepts of the institutional reality to norms and to environment and the mixing of institutional and application's domain semantics. Having artificial institutions where the institutional reality is decoupled but still properly connected (or connectable) to the norms and to the environment is an open issue in the design of MAS societies.

1.2 RESEARCH QUESTIONS

From the described problems, a question to answer is: *how to represent the institutional reality arising from the environment to base the whole regulation in artificial institutions?*

This broad question gives rises to additional ones, considering the problems explained in Section 1.1. Environment and regulation have different representations. Environment, for example, can contain artifacts, specified in terms of operations and observable properties (RICCI; PIUNTI; VIROLI, 2011), and agents, specified in terms of beliefs, desires, and intentions (BRATMAN; ISRAEL; POLLACK, 1988). Regulation can be expressed, for example, through norms defined as obligations, prohibitions, and permissions. Considering the institutional reality as an element that is neither environmental nor regulative, a question to answer is: *what are the proper abstractions to be used to represent (or the constructs to be used to specify) the elements of the institutional reality?*

The institutional reality somehow arises from the environment composed of elements of different natures. Questions related to this point are: *how are these different natures taken into account within the institutional reality? Is there some difference between the institutional counterpart of agents, events, and other kinds of environmental elements? How are these differences represented and managed within the institutional reality?*

Since the regulation, independent of how it is expressed, is assumed to be based on a representation of the institutional reality, questions to answer are: *how to base norms following different models on the same institutional reality? Can the norms, as currently conceived in the literature, be based on a unified representation of institutional reality?*

1.3 HYPOTHESIS

A possible inspiration to answer these questions is the social reality theory by John Searle (SEARLE, 1995, 2009). He considers that the social behaviour of the individuals is defined in the context of *institutions* rooted in an institutional reality *constituted* from the concrete world. Searle's theory has indeed inspired some works on MAS. But these works usually take inspiration only on a rough notion of “count as”, i.e. facts from the environment have some special meaning (or count as other facts) from the normative perspective.

In Searle's theory, however, “count as” is not just a way to provide different meanings to facts from the concrete world. Rather, it is a mean to build an institutional structure – or “reality” – that supports the regulation of the individuals’ behaviour in a society (that Searle refers as *deontic powers*). Thus, Searle’s theory seems to point a direction to answer the previous questions, as it has a conceptual apparatus to capture the notions of institutions and institutional reality arising from the concrete world to support all the deontic powers that make human societies work.

1.4 OBJECTIVES

The main objective of this thesis is *to develop a model of institutional reality to base the normative regulation in artificial institutions.* To achieve this general objective, the following sub-objectives are considered:

1. To define the suitable abstractions to represent the institutional reality;
2. To define how the environmental elements involved in the regulation are brought to the institutional reality, fitting in the proposed abstractions;
3. To define how the regulation, as proposed in the literature, couples with the proposed representation of institutional reality.

These objectives are in a conceptual level as this thesis aims to propose concepts to help to solve the problems previously described. These concepts aim to contribute to solve practical issues in the MAS design and programming. This thesis aims also to advance in this direction. Thus, the following sub-objectives are also considered:

1. To propose a programming language, following the proposed model, to specify the institutional reality of agent societies;
2. To define the architecture of an interpreter for the proposed language;
3. To define the required machinery to base normative regulation following different normative models in the proposed interpreter.

1.5 MOTIVATIONS

As already discussed, MAS societies has, at least implicitly, an institutional reality in terms of which the regulation is expressed. Having institutions where this notion is explicit by proper representations is important to decouple the notion of institutional reality of the notions of norms and environment. Institutional reality can thus be specified as a building block within MAS institutions, side by side with norms, each one fulfilling a particular purpose.

Through such decoupling, norms do not need to deal with the complexities of the heterogeneous, dynamic, evolving environment. This is not, indeed, the purpose of norms. This complexity is moved to the institutional reality. Norms become thus stable (ALDEWERELD et al., 2010). The already mentioned auction scenario is quite illustrative at this point: the way in which bids are given can change, for example, from utterances to some electronic media, but the norms regulating bids can remain unchanged.

The explicit representation of the institutional reality as composed of elements that embody the notion of constitution, with all its complexity, is important to set common interpretations about how the norms are related to the environment, independent of particular views of the agents. For example, Brito et al. (BRITO et al., 2015b) describe a crisis management scenario where agents collaborate to evacuate zones that can be either *secure* or *insecure*. In such a scenario, in spite of the variety of involved agents, organizations, and norms, it is important that the involved parties have the same interpretation about what situations in the environment mean the zones being *secure* and *insecure*.

From the agents' perspective, an explicit representation of the institutional reality enables them to reason about the norms without knowing in advance all the complexities of the environment. In fact, the contrary is possible: the agents can discover how to properly act in the environment by observing how the concepts used in the norms are constituted from the environment.

1.6 DOCUMENT STRUCTURE

In the following, Chapter 2 presents the theoretical elements required to achieve the goals of this work, including the state of the art. These elements are the basis for the proposed model of artificial institution, called SAI (from Situated Artificial Institution), that considers both norms and institutional reality. Chapter 3 presents the representation and dynamics of institutional reality according to the proposed model. Chapter 4 presents an approach to introduce norms in the proposed model of artificial institution, coupling them with the conceived representation of institutional reality. Chapter 5 presents a proposal of architecture of an institutional platform. Chapters 6 and 7 present application examples that help us to evaluate the proposal of this work. Finally, Chapter 8 summarizes the contributions of this work, pointing also some perspectives of future work.

2 BACKGROUND AND STATE OF THE ART

The role of societies is to allow its members to coexist in a shared environment (DIGNUM, 2004). Human societies, along their evolution, have developed means, that are studied by the social sciences, to enable such coexistence. These means have inspired proposals for the coexistence of the agents in their societies. Research in MAS has adopted (and adapted), in different ways, concepts such as *norms*, *institutions*, etc.

Among all the social aspects involved in agent societies, this chapter focuses on the relevant ones to this thesis.[1] More precisely, it looks for the foundations to conceive artificial institutions where the whole regulation is based on the institutional reality, that arises from the environment. Keeping in mind the problems and questions posed in sections 1.1 and 1.2, relevant aspects of Searle's theory on institutions are described in Section 2.1. Moving to the MAS field, Section 2.2 describes how regulation and institutional reality are, in different ways, addressed by works on this kind of system. Conciliating regulation and institutional reality has some open issues in the current state of the art, as shown along Section 2.2.2 and discussed in Section 2.3.

2.1 INSTITUTIONS ACCORDING TO JOHN SEARLE

Institutional reality is part of the broader concept of *social reality*. These concepts are part of a theory proposed by the philosopher John Searle (SEARLE, 1995, 2009). The main fact puzzling Searle is the existence of facts in the world that are objective but that exist only because we believe them to exist. For instance, *money* and *president*: it is possible to objectively state whether or not some piece of paper is money and whether somebody is the president. The fact of some piece of paper being money and somebody being the president, however, is not due to the nature of these elements but due to some social agreement about that. These are *social facts*, contrasting with the *brute facts*, that are those facts that can be explained by the basic sciences and do not depend on any mental attitude from the individuals. For example, the water being composed of hydrogen and oxygen is a brute fact.

Social facts compose the *social reality*. They consist of those facts that require collective intentionality to exist. A social fact exists when the involved parties share intentional states such as believes, desires, and inten-

---

[1] Additional information on social aspects of MAS can be found in (BOELLA; TORRE; VERHAGEN, 2007; BOELLA et al., 2009; ANDRIGHETTO et al., 2013) and in the COIN series of workshops (`http://www.pcs.usp.br/~coin/`).

tions (SEARLE, 1995, page 24). A subclass of social facts are the *institutional facts*, that compose the *institutional reality* as part of the social one. Institutional facts involve *institutions*. For example, two people playing a piece of music just for fun is a social fact while two people playing a piece of music in an orchestra is an institutional fact because they are under an institution (the orchestra, in this case).

There are many definitions of *institutions*. They can be seen as structures that govern the individuals' behaviour in specific contexts, leading them, despite their different opinions, desires, skills, etc. to properly act in compliance with the social expectations (MILLER, 2012). Some classical definitions conceive institutions as constraining entities that define, through elements such as norms and rules, how the individuals should behave (NORTH, 1994; CRAWFORD; OSTROM, 1995). But institutions not just constrain the behaviour of the individuals. They also enable the structure, referred by Searle as *institutional reality*, that makes this constraining possible (COMMONS, 1934; HODGSON, 2006). This is the notion of institution adopted in this work. According to Searle, human institutions are based on the following elements:

1. **Status functions.** In human societies, functions are assigned to objects, people, events, or any other sort of element, that these elements cannot perform solely due to their natural virtues. Rather, these functions are performed because the elements performing them have a recognized status in the society. For example, a piece of paper may have the status (and the consequent function) of "five dollar bill", an individual may have the status (and the consequent function) of "professor", and the raising of hands may have the status (and the consequent function) of "vote" in an election. These status that assign functions to the elements of the concrete world, such as *five dollar bill*, *professor*, and *vote*, are called *status functions*.

2. **Collective agreement.** The notion of status function raises a question: how can the physical elements carry these functions that are not inherent to them? The answer is the existence of the *collective agreement*. For example, a piece of paper is a five dollar bill, an individual is a professor, and the raising of hands is a vote because people agree about that (or at least accept that). Otherwise, they would be just a piece of paper, an individual, and an act of raising of hands.

3. **Deontic powers.** An important element to the functioning of societies is what Searle calls *deontic powers*, analogous to *norms* in MAS field. Roughly, they define what people are expected to do or to avoid. In human societies, norms are expressed through concepts as "rights",

"duties", "obligations", "permissions", "authorizations", etc. In MAS, these concepts are usually subsumed as permissions, obligations, and prohibitions. Examples of norms are (i) "a PhD student is obliged to finish his thesis on time", (ii) "a PhD student is permitted to finish his thesis before the final time", and (iii) "a PhD student is prohibited to committing any plagiarism in his thesis".

4. **Action independent of desires and of physical constraints.** Norms and status functions compose a powerful system that supports human societies. According to Searle, they are "*the glue that holds the human civilization together*" (SEARLE, 2009, page 9). This is because they provide reasons to people act independent of their own desires and independent of physical constraints implemented in the brute reality. Norms are attached to status functions instead of being attached to the concrete elements of the brute reality. For instance, a high wall can be a physical constraint to prevent people to enter in a private property. But if the wall is replaced by a small line of stones meaning the boundary of the property, and a norm is issued stating that individuals are prohibited to enter in private properties, then people have a reason to not cross the small line of stones, entering in the private property, even if they are physically able to do that.

5. **Constitutive rules.** The *constitutive rules* define how the status functions are constituted, connecting them to the brute reality. The *constitutive rules* have the form *X counts as Y in C*, meaning that the element X carries the status function Y in the context C. Recalling the previous example, the line of stones (X) counts as the boundary of the private property (Y) if the property is registered to some owner (C).

From these elements, it is possible to define *institution*. According to Searle, *institutions are systems of constitutive rules and the constitutive rules create the possibility of institutional facts* (SEARLE, 2009, page 10). For example, a constitutive rule might enable the fact of *bob* counting as *PhD student*. Since constitutive rules define constitution of status functions, the building blocks of the institutional reality, composed of institutional facts, are the constituted status functions.[2]

This definition of *institutions* as systems of constitutive rules may, at first glance, hide the regulative character of the institutions. But it is in fact

---

[2] Put in this way, the institutional reality is described in a *fact mode*. Sometimes Searle describes the same institutional reality in an *object mode*, as being composed of institutional *objects* instead of institutional *facts*. In this case, for example, the institutional reality is described as being composed of a *PhD student* constituted by the agent *bob* instead of being composed of the fact of *bob* counting as a *PhD student* (SMITH; SEARLE, 2003).

included in the definition. When Searle claims that institutions are systems of constitutive rules, he means that institutions define some particular interpretation of the brute reality, expressed in terms of constituted status functions, to base the deontic powers. The deontic powers are attached to the status functions. That is to say, the deontic powers, defined through the norms, exist within the institutions only when status functions are constituted. For example, the norm "*a PhD student is obliged to finish his thesis on time*" is effective only when the institution constitutes *PhD student* and *thesis*. Thus, institutions as systems of constitutive rules are those that enable the existence of the institutional reality and that also define the expected individuals' behaviour. Institutions perform these tasks through the following operations:

1. **Creation of status functions.** As status functions are not elements existing by themselves in the world, they need to be "created". At some point of the history of the institutions, people create status functions such as *president*, *money*, *king*, etc.

2. **Constitution of status functions.** There are two types of constitution of status functions:

    (a) **Assignment of a status function Y to a concrete element X.** In this case, a status function is assigned to an existing X (that is to say, X is a concrete element existing in the world independent of any status function assignment). This assignment may be either *specific* or *generic*:

      i. When a status function Y is assigned to a generic X, any element satisfying a set of conditions is the X carrying Y in a context C. For example, "for all x, x is the strongest man in the group, then x is the X that counts as the leader (Y) in any circumstance (C)".

      ii. When a status function Y is assigned to a specific X, the assignment is defined directly to the element X that must carry the status function. An example of this kind of assignment is "*Bob* is the X that counts as the leader of the group (Y) in any circumstance (C)".

    (b) **Freestanding assignment.** In the previous cases, the status function is assigned to a pre-existing element. Searle observes, however, that some status functions are not assigned to any element. They just start to exist in the institution, being created *by fiat*. A famous example mentioned in the literature is the concept of *corporation*: it does not consists only of the people composing it, nor of the buildings where it is placed, nor by its equipments.

All of these elements compose the company and can even change (e.g. people can leave the company) but the *company* still remains. Other example is the virtual money: nowadays financial operations occur without exchanging any physical element. The exchanged money, in this case, simply exists (SEARLE, 2009, page 20). The *freestanding Y* is acknowledged and discussed by authors other than Searle and this is still an open issue in the philosophy (HINDRIKS, 2012; SMITH; SEARLE, 2003; SMITH, 2003).

3. **Power creation.** The power creation relates deontic powers to status functions defining, for example, that the president has some obligations that do not stand to the other individuals or the owner of a 100 dollar bill can buy a good that costs 100 dollars.

The constitution of status function imposes an institutional status on the environmental elements. The power creation assigns institutional functions to the environmental elements that carry the status functions as it defines how they affect the expected individuals' behaviour, expressed through deontic powers.

2.2 INSTITUTIONS IN MAS

According to Searle's theory, human institutions have a regulative dimension (the deontic powers) that is contextualized within the institutional reality. Regulation and institutional reality are addressed – but not necessarily connected to each other – by works on agents' societies. Sections 2.2.1 and 2.2.2 provide, respectively, an overview about how regulation and institutional reality are conceived in MAS.

2.2.1 Regulation in MAS

Norms are the most usual way to express the regulative requirements in MAS (LÓPEZ; LUCK; D'INVERNO, 2002). For this reason, the main focus of this section is on describing how regulation through norms is conceived in MAS (Section 2.2.1.1). After, other ways to express the regulative requirements are presented (Section 2.2.1.2).

2.2.1.1 Norms

A considerable amount of work has been developed addressing norms with different perspectives. Some works are concerned with the logics of norms, other are concerned with the normative representation, specification, monitoring, reasoning, etc. Covering all of these perspectives on norms is beyond the scope of this work. In this thesis, the main point to consider is that norms can express the expected individuals' behaviour in different ways. For example, some normative models consider that the expected individuals' behaviour is expressed only through obligations, while others consider prohibitions or permissions. Normative models can also differ in the way to express the conditions under which the norms must be followed or in how the non compliance with the norms is handled.

This section describes how the different aspects involved in the normative regulation are considered in some of the different normative models described in the literature. To analyse the different normative models under the same perspective, we take as parameter the ADICO model (CRAWFORD; OSTROM, 1995). According to this model, the expected individuals' behaviour can be expressed in terms of ***A***ttributes, ***D***eontic operators, a***I***ms, enforcement ***C***onditions and consequences of the non-compliance (***O***r-else). The next paragraphs explain how these elements are considered in the analysed normative models. The results of our analysis are summarized in Table 1.

The ***Attribute*** defines who are the individuals whose behaviour is regulated by the norm. Some models consider that norms are specified directed to the specific agents acting in the system (e.g. the agent *bob* is obliged to bid). Other models consider that norms are specified directed to *roles* (e.g. agents playing the role *bidder* are obliged to bid). Norms can also be specified to agents playing roles (e.g. the agent *bob*, when playing the role *bidder*, is obliged to bid). In this case, other agents playing the same role are not targeted by the obligation and, in other hand, the same agent is not targeted by the obligation when he is not playing the specified role. Finally, norms can be directed to groups of agents, independent of their roles.

Norms have an associated ***Deontic*** type that defines the expected attitude with respect to some outcome. All the analysed normative models consider norms expressing obligations, defining thus that the agents are obliged to produce an outcome. Some models consider also prohibitions and a lower amount considers also permissions.

The ***aIm*** expresses the outcome that is expected when the norm is

		2OPL[a]	Lopez et al.[b]	(PACHECO, 2012)	Panagiotidi et al.[c]	ISLANDER[d]	$\mathcal{M}$oise^{INST}[e]	ANTE[f]	NormML[g]	OCeAN[h]	NPL[i]
Attribute	Agent		✓						✓		✓
	Role	✓		✓	✓	✓	✓	✓	✓	✓	
	Agents X Roles								✓		
	Groups								✓		
Deontic	Obligation	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Prohibition	✓	✓	✓		✓	✓		✓	✓	
	Permission		✓	✓		✓	✓		✓		
a**I**m	Events		✓	✓		✓		✓	✓	✓	
	States	✓	✓	✓	✓	✓	✓				✓
Condition	Instantiation		✓							✓	
	Activation	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Deadline	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Removal					✓				✓	
	Deactivation		✓								
	Context								✓		
	Maintenance				✓						
	Reparation				✓						
Or-Else	Inst. Feedback	✓				✓		✓	✓		
	Norm chaining	✓	✓	✓		✓	✓	✓	✓	✓	✓

Tabela 1 – Analysis of normative models

---

[a] (DASTANI et al., 2009; TINNEMEIER et al., 2009; ALECHINA et al., 2013)

[b] (LOPEZ; LUCK, 2002; LÓPEZ; LUCK, 2003)

[c] (PANAGIOTIDI; ÁLVAREZ-NAPAGAO; VÁZQUEZ-SALCEDA, 2013)

[d] (VÁZQUEZ-SALCEDA; ALDEWERELD; DIGNUM, 2004a, 2004b)

[e] (GATEAU, 2007)

[f] (CARDOSO; OLIVEIRA, 2007, 2008)

[g] (FIGUEIREDO; SILVA; BRAGA, 2011)

[h] (FORNARA; COLOMBETTI, 2006, 2009)

[i] (HÜBNER; BOISSIER; BORDINI, 2010, 2011)

followed. As noted in (VOS; BALKE; SATOH, 2013), norms can define either states to be achieved or events to be produced by the agents. Models combining both of them are mainly conceptual, focusing on how norms are expressed and ignoring the differences between the monitoring of events and states. Norms based on states have a "see to it approach": agents must see to them that some state of affairs holds, no matter how it is achieved (MARRA; KLEIN, 2015). In other hand, norms based on events usually perform their monitoring in an individual agent level: the specified event must be produced by an specific agent.

The ***Conditions*** capture the circumstances under which the norm must (or does not need to) be followed. At this point it is important to introduce the notion of *norm instance*. Norms exist in the society and the agents work with norm instances (LÓPEZ; LUCK, 2003), that are a kind of *copy* of the original norm (grounding existing variables). It is possible the existence of many instances related to the same norm (e.g. an instance for each targeted agent). For example, the norm "*buyers are obliged to pay*" could produce the instances "*bob is obliged to pay $100*" and "*tom is obliged to pay $50*". *Conditions* refer to the circumstances that change the lifecycle of a norm instance. Lifecycles may be different according to the normative model. An example of lifecycle is illustrated in Figure 2. Most of the approaches consider that norms have an *activation* condition whose satisfaction determines the creation of norm instances. But norms can have a particular *instantiation* condition, whose satisfaction determines the creation of instances, that remains inactive (sort of latency state) until activation conditions are satisfied (LOPEZ; LUCK, 2002). In all the analysed models, the *deadline* condition determines the limit before which the norm must be fulfilled. In the case of obligations, the aim must be done before the deadline. In the case of prohibitions, the aim must be avoided until the deadline. While some models consider that satisfied and violated instances cease to exist, other models consider that these instances, for some reason, remain existing, being deleted only when an explicit *removal* condition is satisfied. Some models consider that, before being removed, violations can be repaired by satisfying a *reparation* condition. It is possible also to consider a *deactivation* condition that expresses circumstances under which the instance remains existing but do not need to be followed. *Deactivation*, in this case, is a sort of "pause" in the effects of the norm instance. If the instance is active and the *deactivation* is no longer satisfied, then the instance must be followed again. Some models have an implicit deactivation: the instance is deactivated when the activation condition is no longer satisfied. The activation condition, in this case, is also a maintenance condition. But it is possible also to consider an explicit maintenance condition. In this case, the instance remains active while the maintenance condition is satisfied, even the activation condition ceases to hold. In this case, the instance is not violated. Rather it can be repaired by satisfying the *reparation* condition. There is a model also considering a *context* that can be seen as the dual of the *deactivation* condition. An active instance must be followed only by the agents acting in a specific context.

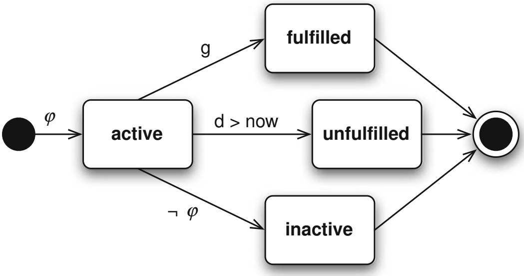


Figura 2 – Lifecycle of a NPL norm instance (HÜBNER; BOISSIER; BORDINI, 2011): a norm instance is created in the *active* state when the activation condition φ is satisfied; it is *fulfilled* when the goal g is achieved; it becomes *inactive* if the activation condition φ ceases to hold; and it is considered *unfulfilled* if the deadline d is achieved.

It is important to remark that the *aim* could be considered a "satisfaction" condition. But it is not described here among the *conditions* because the ADICO model considers *aim* and *conditions* as different components of the norms. A second remark is that although different models consider some common conditions, these conditions can be affect the lifecycle of norm instances in different ways, according to the normative model. For example, while the model of (PANAGIOTIDI; ÁLVAREZ-NAPAGAO; VÁZQUEZ-SALCEDA, 2013) considers the satisfaction of the *activation* condition only creates the norm instances, the NPL model considers that this condition also determines the maintenance of the norm instance. This brief description of the norm conditions aims to provide a global view of the conditions that change the normative state instead of to describe the detailed semantics of the different normative models.

The ***Or**-else* element of a norm defines the consequences of non compliance, i.e. the sanctions to be applied to the misbehaving agents. A first approach to sanctions is the *institutional feedback*: the normative platform imposes some change in the environment. The normative platform has access to the environment to change it. The sanction, in this case, can be specified as part of the norm. Other possible approach is the *norm chaining*: a violation activates a new norm instance. The sanction, in this case, can be specified as another norm. Some agent – the one that has violated the norm or someone else – has a new norm to follow. This new norm instance can be also violated triggering a new norm and so on. The chaining stops when a violation does not trigger a new norm. This is a decision of the designer of the norms (LÓPEZ; LUCK; D'INVERNO, 2007).

2.2.1.2 Other regulative representations

Although norms are the most usual way to express the expected agents' behaviour, there are other representations described in the literature with this same purpose. One of these representations is the *commitment*, that expresses the expected agents' behaviour in a particular kind of activity, that is the interaction with other agents. While norms usually refer only to the bearer (or the *attribute*), commitments involve two agents: the *debtor*, that is expected to have some behaviour, and the *creditor*, that expects that behaviour from the debtor. Similar to norms, commitments can be satisfied, violated, etc (DASTANI; TORRE; YORKE-SMITH, 2012). Commitments and norms are sometimes deemed as interchangeable concepts (LÓPEZ; LUCK; D'INVERNO, 2004). The relation between commitments and norms is discussed in (SINGH, 1999), (CARDOSO; OLIVEIRA, 2007), and (FORNARA et al., 2008).

The regulation of the agents' behaviour can also be expressed through *organizations*, that provide more complex guidelines to the agents' actions. Organizations define, for example, the proper sequence of the activities to be performed, and the social structures, in terms of groups, required to fulfil the social goals. Some known organizational models are OperA (DIGNUM, 2004), TÆMS (DECKER, 1996; LESSER et al., 2004), $\mathcal{M}$OISE (HÜBNER; SICHMAN; BOISSIER, 2002), PopOrg (DEMAZEAU; COSTA, 1996; COSTA; DIMURO, 2007, 2009), and AGR (FERBER; GUTKNECHT; MICHEL, 2004; FERBER; MICHEL; BÁEZ-BARRANCO, 2004). Although organizations are more complex ways to describe how the agents are expected to behave, they can sometimes be translated to a set of norms, as observed in (HÜBNER; BOISSIER; BORDINI, 2010, 2011).

Norms and other regulative representations express, in the agents' societies, what Searle calls "deontic powers". These representations usually abstract from the concrete environment where the agents act (ALDEWERELD et al., 2010), being compliant with – even not explicitly inspired by – the Searle's assumption of deontic powers contextualized within an institutional reality decoupled of the concrete world. The next section explains how current works address this institutional reality in MAS.

2.2.2 State of the art on institutional reality in MAS

The main idea behind the notion of constitution – facts from the environment counting as another facts related to the regulation – has inspired works in MAS. In different ways, these works consider that the *count as* relation defined by the constitutive rules produces elements to be considered in the regulation of the systems. We consider these elements as the components of the *institutional reality* existing in MAS, even this notion is not usually explicit in works on *count as* applied to these systems. This section analyses how the works on MAS conceive the institutional reality built through the count as relation. Some of the existing approaches are analysed from the perspective of the problems described in Section 1.1. Thus, we describe (i) how the institutional reality is represented, (ii) how the regulation is based on these representations, and (iii) how this institutional reality arises from the complex environment where agents interact. Drawbacks and limitations of these aspects are discussed. Inspired by the work presented in (BRITO; HÜBNER, 2014), current works are divided in three groups, presented from sections 2.2.2.2 to 2.2.2.1.

2.2.2.1 Institutional reality as an interoperability issue

Some works propose constitutive rules to specify that the environmental facts count as other kinds of facts but leave to the regulative platform to handle the results of constitution. Constitutive rules are used more to fill an interoperability gap between environmental and regulative platforms than to actually model the institutional reality. Briefly, constitutive rules define that some environmental facts produce (or *count as*) some data to be delivered to the regulative platform, that is in charge to handle the received data, assigning it some meaning to be considered in the regulation. That is to say, the regulative platform is responsible for building the institutional reality based on the received data.

A first example of this approach is the work by (PIUNTI et al., 2010). The constitutive rules, in this proposal, define that environmental events count as the triggering of operations in ORA4MAS artifacts, that are a technological support for the $\mathcal{M}$OISE organizational model (HÜBNER et al., 2009).[3] Figure 3 shows a constitutive rule following this approach. In the application from where it was excerpted, the rule specifies that an object *BilllingMachine* producing the event *pay* counts as the operation *setGoalAchieved* being executed in the artifact *visitorSchBoard*. Another example of this approach is the work by (BRITO; HÜBNER; BORDINI, 2013), where constitutive rules define properties that the regulative representations should have. Figure 4 shows a rule specifying that the event *pay* occurring in the environment counts as the property *goalState(bill_paid,satisfied,Ag)* holding in the normative platform. In the particular application, such property holds when the organizational goal *bill_paid* is satisfied by the agent *Ag*. A third example of this approach is the work by (CAMPOS et al., 2009). In this case, the institutional reality is defined by special agents named *governors*. They observe the facts occurring in the environment defining whether these facts count as something from the regulative perspective.

```
+op_completed("BillingMachine",Ag, pay)
-> apply("visitorSchBoard",setGoalAchieved(Ag, pay_visit)).
```

Figura 3 – Constitutive rule (PIUNTI et al., 2010)

```
+ pay[agent_name(Ag)]
  count-as goalState(bill_paid,satisfied, Ag).
```

Figura 4 – Constitutive rule (BRITO; HÜBNER; BORDINI, 2013).

The *regulative semantics* is not present in the constitutive models and, thus, the regulative consequences of the environmental facts are not explicit in the constitutive rules. Rather, such consequences are defined by the mechanisms that handle the results of the interpretation of the constitutive rules. In (PIUNTI et al., 2010), the mechanism are the ORA4MAS artifacts; in (BRITO; HÜBNER; BORDINI, 2013), it is some *ad hoc* interface placed between the constitutive and regulative engines; in (CAMPOS et al., 2009), the *governors* are this mechanism. For example, the rule shown in Figure 3 expresses that the occurrence of the environmental event *op_completed*, with some parameters, triggers the operation *setGoalAchieved*, with some parameters, in the artifact *visitorSchBoard*. But the intended regulative meaning –

---

[3] Roughly speaking, *artifacts* can be compared to *objects* of the object oriented programming (OOP). The artifacts encapsulate the $\mathcal{M}$OISE abstractions, that are handled through *operations* (comparable to the *methods of OOP*).

the execution of the operation *pay* counting as the achievement of the goal *pay_visit* – is not explicit in the constitutive rule because notions such as *goal achievement* are not part of the constitutive model. This meaning is in the mind of the designer and, in the practical application, it is given by the artifact that, through the operation *setGoalAchieved*, sets the goal as achieved. Notice that *setGoalAchieved* is a parameter of the operator *apply* instead of being a constructor of the constitutive language. Similarly, considering the constitutive rule shown in Figure 4, the designer may know that, in a particular application, the property *goalState(bill_paid,satisfied, Ag)* holds when the goal *bill_paid* is achieved. But such meaning is not explicit in the constitutive rule because the notion of *goal achievement* is not part of the model proposed by (BRITO; HÜBNER; BORDINI, 2013).

As the regulative semantics is not explicit in the constitutive rules, to reason about the regulative consequences of the environmental facts, the agents need to know how the regulative platforms handle the results of the interpretation of the constitutive rules. In the case of (PIUNTI et al., 2010), the agents must know the functioning of the ORA4MAS artifacts; in the case of (BRITO; HÜBNER; BORDINI, 2013), they must know how the specific interfaces handle the results of the constitution; in the case of (CAMPOS et al., 2009), they must know how the *governors* interpret the facts from the environment. Furthermore, as there is not any link between the regulative semantics and the constitutive rules, it is possible to specify that environmental facts count as “anything”. As shown in Figure 5, it is possible to specify rules that are syntactically and semantically correct but that have no effect in the regulation.

```
+op_completed("BillingMachine",Ag, pay)
-> apply("visitorSchBoard",meaninglessParam(Ag, pay_visit)).
```

Figura 5 – Meaningless code excerpt following the model of (PIUNTI et al., 2010)

Despite the aforementioned limitations, this approach can be considered as having a *wide regulative support*. The constitutive models do not impose, themselves, limitations regarding the related regulative models. Such support depends only on the regulative platforms being able to handle the results of constitution. In the work of (CAMPOS et al., 2009), the regulative support depends on the skills of the *governors* collect the facts from the environment and handle regulative representations accordingly. In the work of (BRITO; HÜBNER; BORDINI, 2013), any regulative platform having proper interfaces can handle the results of the constitutive rules. Although the work of (PIUNTI et al., 2010) is conceived in the context of ORA4MAS/$\mathcal{M}$OISE, the proposed constitutive rules can be applied to any regulative model embedded in CArtAgO artifacts.

In this approach it is also not possible to ensure consistency between the environment and regulation as it is possible to define the same environmental fact as producing conflicting results in the regulation. For example, the same program containing the constitutive rule shown in Figure 4 could also contain a rule defining that the occurrence of the event *pay* counts as the goal *bill_paid* being unsatisfied. This is because the meaning of results of constitution are not part of the constitutive model and, thus, opposite conditions cannot be checked.

2.2.2.2 Institutional reality as normative state

Some works consider the institutional reality as being the normative state, that is the state of the norms and other abstractions regulating the system. In this case, for example, facts occurring in the environment count as norm *activations*, *violations*, *fulfilments*, etc. Works related to the 2OPL normative model go in this direction (DASTANI et al., 2009). They consider that environmental facts may count specifically as norm violations. The constitutive rule below follows this approach to define that the environment satisfying the formula in the left side counts as a norm violation of the type *viol_size*:

$$received(As) \wedge member((A,Id),As) \wedge pages(Id) > 15 \Longrightarrow viol\_size(A)$$

Another example of this approach is the proposal of (DASTANI; TORRE; YORKE-SMITH, 2012), that considers that the facts composing the institutional reality are the possible states of commitments, whose lifecycle is shown in Figure 6. The environmental facts, in this proposal, may count as commitments being in the states *conditional* (that is the initial state), *expired*, *detached*, *terminated*, *satisfied*, or *violated*. Constitutive rules following this approach are illustrated in Figure 7. The line 1 defines that the offering of an agent x to an agent y to perform q before the instant d_2, conditioned to the performance of p by y before d_1, counts as a commitment from x to y with respect to q being in the state *conditional* (C^{c}). The line 2 specifies that the agent x informing to y that he has been done q before d_1 counts as the satisfaction of the commitment, that moves from the state *conditional* to *satisfied* (C^{s}).

In the mentioned works, the *regulative semantics* is considered in the

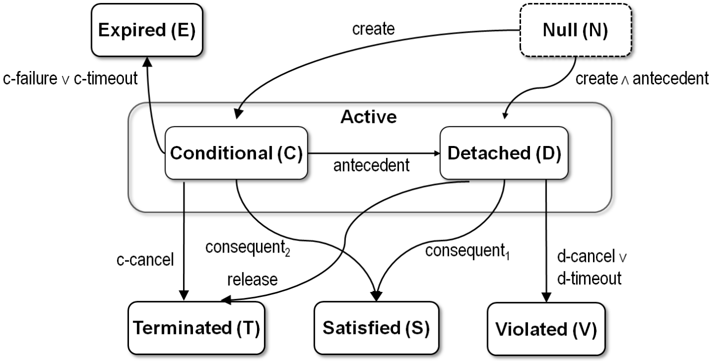


Figura 6 – Lifecycle of a commitment (DASTANI; TORRE; YORKE-SMITH, 2012)

1 $offer(x,y,p,q,d_1,d_2) \Longrightarrow_{cr} \mathsf{C}^c(x,y,p,q,d_1,d_2)$

2 $tell(x,y,q) \wedge \mathsf{C}^c(x,y,p,q,d_1,d_2) \wedge \neg d1 \wedge q \Longrightarrow_{cr} \mathsf{C}^s(x,y,p,q,d_1,d_2)$

Figura 7 – Constitutive rules for commitments (DASTANI; TORRE; YORKE-SMITH, 2012).

constitutive model. The results of the constitution (*norm violations*, *conditional commitments*, etc.) are explicitly related to the semantics of the regulative abstractions. The regulative consequences of the facts occurring in the environment are thus explicit in the constitutive rules. For example, it is explicit in the constitutive rule 1 of Figure 7 that the utterance of an *offer* produces a commitment in the *conditional* state. The notion of *conditional commitment*, as well as the other envisaged commitment states, is part of both the regulative and the constitutive models. This is an advantage, as the designer of the constitutive rules can only specify consequences for the environmental facts that make sense from the regulative point of view. For example, again considering the work by (DASTANI; TORRE; YORKE-SMITH, 2012), a constitutive rule specifying that an environmental fact counts as something different of a commitment *created*, *expired*, *detached*, *terminated*, *satisfied*, or *violated* (denoted respectively by C^{c}, C^{e}, C^{d}, C^{t}, C^{s}, and C^{v}) is syntactically and semantically wrong. The elements C^{c}, C^{e}, C^{d}, C^{t}, C^{s}, and C^{v}, that are related to the semantics of commitments, are constructs of the constitutive language. The constitutive rules are thus always consistent with the regulation.

However, such mixing of the semantics of constitution and regulation limits the institutional reality to base a specific kind of regulative abstraction. The institutional reality produced by the constitutive rules of (DASTANI; TORRE; YORKE-SMITH, 2012) cannot base commitments whose lifecycle contains states others than those shown in Figure 6. Similarly, The institutional reality produced by the constitutive rules of (DASTANI et al., 2009) cannot base regulative abstractions whose lifecycle contains states others than those envisaged by the considered normative model. This approach has, thus, a *limited regulative support*.

An institutional reality composed of facts related to the regulative dynamics does not capture the grounding in the environment of the concepts referred by the norms. This may lead to inconsistencies as the same environmental fact may count as conflicting outcomes in the regulation. For example, one could specify that the same environmental fact counts as the activation of both the norms "*bidder is obliged to bid while the auction is open*" and "*bidder is prohibited to bid while the auction is not open*" even if the norms are declared to be active under opposite conditions.[4] The concept "*open auction*" is not linked to the environment and thus these opposite conditions cannot be checked with respect to the actual environment. Furthermore, when the concepts used in the norms are not part of the institutional reality, the agents can reason about the regulative consequences of their actions but it might be dif-

---

[4] A discussion on normative conflict can be found in (VASCONCELOS; KOLLINGBAUM; NORMAN, 2009).

ficult to them to reason about the institutional meaning of their actions. For example, consider that, in spite of the aforementioned norms, an agent do not want to bid. If a constitutive rule defines that "*raising the hands counts as to bid*", the agent knows that the action of raising the hands fulfils the norm but it cannot know that such action conflicts with their personal will of not to bid.[5]

2.2.2.3 Institutional reality as constitutive state

Some works conceive the institutional reality as being composed of institutional counterparts of the environmental elements. Constitutive rules can define, for example, that some agents count as *bidder* and some actions count as *bid*. The standing relations between environmental elements and their institutional counterparts, defined by the constitutive rules, compose what we refer as the *constitutive state* of the institution. Even the normative regulation ranges over the elements in the environment, norms are supposed to be specified referring to their institutional counterparts. For example, the norm "*bidder is obliged to bid*" defines an obligation for the agents that count as *bidder* to perform actions that count as *bid*.

Although proposals following this approach consider specific normative models, their general idea – producing a constitutive state whose elements are referred by the norms – could be applied to different normative models that base the regulation on concepts that do not belong to the environment. The institutional reality, in this case, provides a *wide regulative support*.

Works following this approach have different conceptions about the institutional counterparts of the environmental elements. In the case of (ALDEWERELD et al., 2010; BOELLA; TORRE, 2004, 2004, 2006b, 2006a), these institutional counterparts are words to be used in the normative specification. These words, however, are not typed, i.e. there is not a taxonomy classifying the elements in the institutional reality. This feature raises issues to relate the institutional reality both with the norms and with the environment:

- Regarding the environment, the words do not capture the nature of the environmental elements under regulation. This makes room for problems in the specification and in the management of the institutional reality. For example, *bidder* and *bid* can be deemed, intuitively, as the

---

[5] This kind of conflict can be seen as a conflict between norms and values. A discussion on this topic can be found in (FIGUEIREDO; SILVA, 2013).

counterparts, within the institution, of agents acting and events occurring in the environment. But if the elements of the institutional reality are just words, such semantics is not explicit. One could, in this case, specify that agents count a *bid* and their actions count as *bidder*. An explicit relation between the elements of the institutional reality and the nature of their environmental counterparts is also important in the management of the institutional reality. Elements corresponding to agents (e.g. *bidder*) should exist in the institutional reality only while the agents are participating in the system. Elements corresponding to events (e.g. *bid*) should exist only while the event is occurring. This applies to all the kinds of elements possibly composing the institutional reality.

- Regarding the norms, it is not possible to establish an explicit relation between the components of the norms and the elements of the institutional reality. For example, the institutional reality can be composed of *bidder* and *bid* but it is not explicit whether these elements can be used as *attribute*, *aim*, *conditions*, or *or-else* components of the norms. One could specify both the norms "*bidder is obliged to bid*" and "*bid is obliged to bidder*", even the second one does not make sense. Establishing proper relations between norms and institutional reality, in this case, requires to know both the meaning of the concepts belonging to the institutional reality in the applications' domain and the semantics of the normative model (e.g. *bidder* can be the *attribute* of a norm because, in the application, it corresponds to an agent).

In a slightly different direction, the works by (FORNARA et al., 2008), (VIGANÒ; COLOMBETTI, 2007, 2008), (CLIFFE; VOS; PADGET, 2007), and (CARDOSO; OLIVEIRA, 2007) consider that the institutional reality is composed of *institutional events*, that are the counterpart of events occurring in the environment. Norms can thus refer to the elements of the institutional reality considering this semantics, that is also considered in the management of the institutional reality. But, as noted by (VOS; BALKE; SATOH, 2013), the regulation of MAS should be based on events but also on states. If the institutional reality is composed only of institutional events, either the regulation is exclusively based on events or the norms must refer directly to the environmental states, loosing the advantages of being based on the institutional reality. In this case, the regulative semantics is partially considered in the constitutive rules and the consistency between regulation and environment is also partial (stands only with respect to the events). The institutional reality produced by these works does not have a wide regulative support: it supports only normative models exclusively based on events.

2.3 CONCLUSIONS

To conclude this chapter, we first recall the goal of this thesis, that is to model artificial institutions that consider not only the regulative requirements but also the institutional reality where the regulation is based on. Regulation is a well addressed topic in MAS. There are many proposed ways to represent the expected agents' behaviour, as described in Section 2.2.1. Institutional reality, on its turn, is a less addressed topic. Current proposals present drawbacks such as limited regulative support (the institutional reality can base a limited set of regulative representations) and lack of regulative semantics (the impact of the results of constitution in the regulation is not explicit in the constitutive rules). These drawbacks raise issues such as problems to the agents to reason about the regulative consequences of their actions, constitutive languages prone to errors, inconsistencies between regulation and environment, and conflicting regulative outcomes.

The drawbacks observed in the current approaches for institutional reality seem to be related to the fact the current proposals take a rough inspiration in the notion of *count as*. They consider it as a way to provide some special meaning, to be considered in the regulation, to the environmental facts. From Searle's theory, however, we can see that *count as* is more than a way to provide different meanings to brute facts. It is a mean to operate *constitution*, building the institutional reality that supports the regulation of the individuals' behaviour in a society. Such "building" provided by the constitution involves proper building blocks and operations, as described in Section 2.1. Through these building blocks and operations, human institutions both represent the institutional reality and base the regulation of individuals' activities in the societies. These building blocks and operations inspire the constitutive model introduced in the next chapter.

3 CONSTITUTION: BUILDING THE INSTITUTIONAL REALITY IN ARTIFICIAL INSTITUTIONS

The previous chapter explained what an institution is according to Searle's theory. Resuming that definition, institutions are "systems of constitutive rules". Such a system provides, through the constitution of the status functions, an interpretation of the brute reality, enabling the existence of the institutional one. The expected individuals' behaviour is contextualized within this institutional reality. For example, in an institutional reality composed of *buyers* and *payments*, the norm "*buyers are obliged to pay for their purchases*" stands to all the individuals considered by the institution as *buyers* and it is fulfilled through actions considered by the institution as *payment*.

This conception of institution inspires the model of artificial institution proposed in this thesis, that considers both the regulative requirements (specified through norms) and the institutional reality where they are contextualized (specified through constitutive rules). Such an institution is *situated* in the environment as there is a clear correspondence between specific environmental circumstances and the existence of the elements in the institutional reality. For this reason, we call the proposed model as SAI (from *Situated Artificial Institutions*).[1] This chapter presents a model of institutional reality to be considered in the SAI model. Part of this content is also presented in (BRITO; HÜBNER; BOISSIER, 2014, 2015a). While SAI introduces a representation of institutional reality, as presented in this chapter, the regulation is supposed to be provided by norms without considering a particular normative model. The main novelty with respect to norms in SAI is their coupling with the institutional reality. This topic is addressed in Chapter 4. Regarding the objectives of this thesis, stated in Section 1.4, this chapter (i) defines the abstractions to represent the institutional reality, (ii) defines how the environmental elements involved in the regulation are brought to the institutional reality, fitting in the proposed abstractions, and (iii) proposes a programming language, following the proposed model, to specify the institutional reality in agent societies. The questions to be answered in this chapter are:

1. What are the proper abstractions to be used to represent (or the constructs to be used to specify) the elements of the institutional reality?
2. Is there some difference between the institutional counterpart of agents, events, and other kinds of environmental elements?

---

[1] The name *Artificial Institution* is proposed by (FORNARA et al., 2008) to refer to an extension to the concept of *electronic institution* (ESTEVA et al., 2001). While electronic institutions focus on norms, artificial institutions contain additional elements to represent other social aspects of MAS.

3. How are these differences represented and managed within the institutional reality?

In the following, Section 3.1 presents an overview of the institutional reality in SAI. Section 3.2 introduces the representations used to define how the institutional reality is created from the environment where the agents act. Section 3.3 defines the dynamics of the institutional reality. Section 3.4 illustrates the evolution of the constitutive state, that is the representation of the institutional reality in SAI, based on the environmental dynamics.

3.1 INTRODUCTION TO INSTITUTIONAL REALITY IN SAI

SAI assumes that norms refer to elements belonging to the institutional reality of the system instead of referring directly to elements belonging to the environment. For example, the norm "*the winner of an auction is obliged to pay its offer, otherwise it is fined*" makes sense in an auction scenario. The norm, however, does not specify aspects such as (i) what an agent should do to become the *winner* of the auction, (ii) what an agent must do to perform the *payment*, or (iii) how the *fine* is applied. In this scenario, *winner*, *payment*, and *fine* are elements of the institutional reality enabled by the institution. The institution is in charge of defining when an agent is considered a *winner*, when an action is considered a *payment*, and what must be done to apply a *fine*.

The abstract elements referred by the norms (such as *winner*, *payment*, and *fine* in the previous example) can be seen as *status functions*: they are status, assigned by the institution to the environmental elements, that impose functions to these elements. For instance, agents may have the status (and the correspondent function) of *winner*, the action of exchanging paper bills may have the status (and the corresponding function) of *payment*, etc.[2] In Searle's theory, there is not a hard concern about the kind of element that can carry a status function. He claims that status functions are "*performed by an object(s), person(s), or other sort of entity(ies)*"(SEARLE, 2009, page 94). SAI, for being a more formal model, limits the sorts of entities that can carry status functions. More precisely, it considers that status functions can be assigned to agents acting, events occurring, and states holding in the environment. For example, in a certain institution:

- An agent may have the status function of auctioneer. But it has such

---

[2] In this thesis, as in Searles' work, the expression "*status function*" subsumes both the *status* and the corresponding *function* assigned by the institution to the environmental elements. For example, the agent *bob* carrying the status function *bidder* means that *bob* has both the status of *bidder* and the functions corresponding to such status.

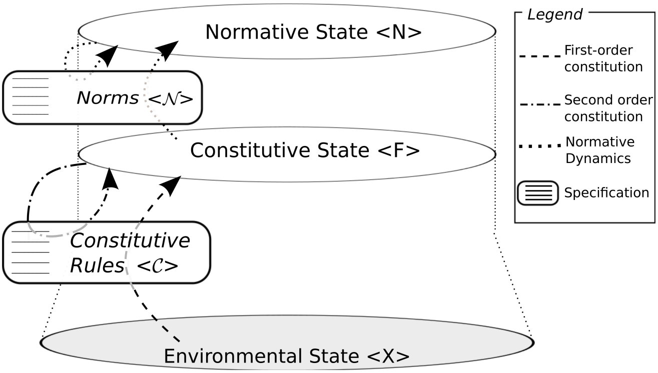
status function due to an institutional assignment. The agent may be implemented with expertise to be an auctioneer and may intend to be an auctioneer, but without the institutional assignment of the status function *auctioneer*, it will not be considered at the institutional level as playing that function.

- The event corresponding to the utterance of "I offer $100,00" may have the status function of "bid" or "counter-proposal", depending on the institutional assignments.

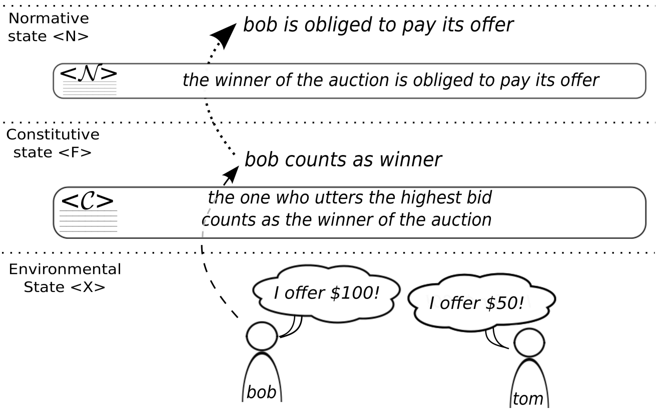
- The state where "more than twenty people are inside a room at Friday 10am" may have the status function, in the institution, the minimum quorum for an auction.

It is assumed that status functions are assigned to these three kinds of elements because they are the kinds of elements involved in the regulation of the MAS. Agents are the autonomous entities whose behaviour is prescribed by the norms. This behaviour is prescribed in terms of occurrence of events and holding of states, as observed by (VOS; BALKE; SATOH, 2013) and analysed in Section 2.2.

The effectiveness of a norm specified through status functions depends on its connection with the environment as its dynamics (activation, fulfilment, etc.) results of facts occurring there. Such a connection is established when the status functions are constituted, according to *constitutive rules*, from the *environmental elements* (Figure 8). The set of constituted status functions is the *constitutive state* of the institution, that is the SAI representation of the institutional reality. Based on the constitutive state, norms are activated, violated, fulfilled, etc., producing the *normative state*, that is the institutional view regarding the expected behaviour of the agents. Consider, for example, an institution where a constitutive rule states that "*the agent that utters the highest bid counts as the winner of the auction*" and a norm states that "*the winner of the auction is obliged to pay its offer*". If the agent *bob* utters the highest bid, then, in the constitutive state, *bob* counts as the winner of the auction and, in the normative state, *bob* is obliged to pay its offer (Figure 8(b)). Section 3.2 describes how constitutive rules specify the constitution of status functions from the environmental elements in SAI. Section 3.3 describes the dynamics of such constitution.



(a) Abstract overview



(b) Scenario overview

Figura 8 – SAI overview: constitutive rules specify how the constitutive state is built from the environmental state while norms specify how the normative state is built from the constitutive state.

3.2 CONSTITUTIVE SPECIFICATION

The constitutive specification designs the institutional reality in SAI. It defines, through constitutive rules, what are the elements composing the constitutive state (i.e. what are the constituted status functions) according to the different possible environmental circumstances. In order to define constitutive rules, we will first introduce the elements to which they refer (environmental elements) and those that participate to their definition (status functions).

Definition 3.2.1 (Environmental elements) *The environmental elements of interest in SAI are represented by* $\mathcal{X} = \mathcal{A}_{\mathcal{X}} \cup \mathcal{E}_{\mathcal{X}} \cup \mathcal{S}_{\mathcal{X}}$ *where* $\mathcal{A}_{\mathcal{X}}$ *is the set of agents possibly acting in the system,* $\mathcal{E}_{\mathcal{X}}$ *is the set of events that may happen in the environment, and* $\mathcal{S}_{\mathcal{X}}$ *is the set of properties used to describe the possible states of the environment.*

We use first-order logic syntax to represent the elements of these sets. Agents in $\mathcal{A}_{\mathcal{X}}$ are represented by constants (e.g. *bob*). Events in $\mathcal{E}_{\mathcal{X}}$ are pairs (e, a) where e is an atomic formula (or simply *atom*) identifying the event (e.g. $offer(100)$) and a is (i) either a constant identifying the agent that has triggered the event or (ii) λ if the event is produced by the environment itself (e.g. a clock tick). Properties in $\mathcal{S}_{\mathcal{X}}$ are represented by atoms.[3]

It is important to observe that the set $\mathcal{X}$ is a representation of the elements that *potentially* take part to the environment.[4] For example, when a SAI specification contains an event $e_{\mathcal{X}} \in \mathcal{E}_{\mathcal{X}}$, it does not mean that $e_{\mathcal{X}}$ has happened in the environment. Rather, it means that the designer of the institution assumes that $e_{\mathcal{X}}$ may happen.

Definition 3.2.2 (Status function) *Status functions are functions that the environmental elements may perform in the institution independent of their design aspects. The status functions of a SAI are represented by* $\mathcal{F} = \mathcal{A}_{\mathcal{F}} \cup \mathcal{E}_{\mathcal{F}} \cup \mathcal{S}_{\mathcal{F}}$ *where* $\mathcal{A}_{\mathcal{F}}$ *is the set of agent-status functions (i.e. status functions assignable to agents* $a_{\mathcal{X}} \in A_{\mathcal{X}}$*),* $\mathcal{E}_{\mathcal{F}}$ *is the set of event-status functions (i.e. status functions assignable to events* $e_{\mathcal{X}} \in E_{\mathcal{X}}$*), and* $\mathcal{S}_{\mathcal{F}}$ *is the set of state-status functions (i.e. status functions assignable to states* $s_{\mathcal{X}} \in S_{\mathcal{X}}$*).*

Agent-status functions are represented by constants. Event- and state status functions are represented by atoms.

---

[3] Precise definitions of *constants* and *atoms* can be found in (BRACHMAN; LEVESQUE, 2004).

[4] It is beyond of the scope of this work to deal in details with the environment. We just consider the elements of $\mathcal{X}$ as existing outside the institution, being available thanks to reliable interfaces.

Definition 3.2.3 (Constitutive rule) *The set of all constitutive rules of a SAI is represented by* $\mathcal{C}$. *A constitutive rule* $c \in \mathcal{C}$ *is a tuple* $\langle x, y, t, m \rangle$ *meaning that* $x \in \mathcal{F} \cup \mathcal{X} \cup \{\lambda\}$ *counts as (i.e. x has the status function)* $y \in \mathcal{F}$ *when the event* $t \in \mathcal{E}_{\mathcal{F}} \cup \mathcal{E}_{\mathcal{X}} \cup \{\lambda\}$ *has happened and while* $m \in W$ *holds.*[5]

In Searle's theory, a constitutive rule expresses that *X counts as Y in the context C*. While in Searle's theory the context C is true when generic circumstances hold, we consider that the context is true (i) `when` some event t has happened and (ii) `while` a certain state expressed by the logical formula m holds. Elements belonging to the set W, as is the case of m, are logical formulae with specific syntax and semantics (cf. expressions 3.1 to 3.3). In the case of $t = \lambda \wedge m = \top$, the constitutive rule is simply read as x `count-as` y since y is assigned to x in any circumstance. When x actually counts as y (i.e. when the conditions t and m declared in the constitutive rule are true), we say that there is a *status function assignment* (SFA) of the status function y to the element x (i.e. y is *assigned* to y). The formal definition of SFA involves the representations of constitutive dynamics, introduced later. For this reason, SFAs are formally defined in Definition 3.3.2. The establishment of a SFA of y to some x is the *constitution* of y. The count-as relation performs the constitution of the status function y as follows:

- *Assignment to an element x.* This kind of constitution applies to rules where $x \neq \lambda$. In this case, the constitutive rules define that a status function y is assigned to an existing element x, that may be either a concrete element belonging to the environment or another status function. For example the rule $\langle bob, bidder, offer(10), auction\_running \rangle$ specifies the assignment of a status function to a concrete element (that we name *first-order constitution*): it means that the agent bob carries the status function of bidder after having uttered its offer and while the auction is running. Constitutive rules can also specify the assignment of a status function to another status function (that we name *second-order constitution*). For example the rule $\langle bidder, auction\_participant, \lambda, \top \rangle$ specifies the assignment of the status function *auction_participant* to the agents that have the status function of *bidder*. First- and second-order constitution are formally defined, respectively, in sections 3.3.2 and 3.3.3.

- *Freestanding assignment.* This kind of constitution applies to rules where $x = \lambda$. In this case, there is not an element that carries the status function. Rather, the constitutive rules just state that the status function exists in a certain context. For example

---

[5] λ represents that the element is not present in the constitutive rule.

the rule $\langle \lambda, auction\_running, \lambda, \neg auction\_finished)\rangle$ means that the property *auction_running* holds in the institution when the property *auction_finished* does not hold. In this case, there is not any property in the environment that carries the status function of *auction_running*. The idea of elements that exist in the institution but do not have a corresponding in the environment is recognized by Searle (SEARLE, 1995, 2009) and by other related authors (HINDRIKS, 2012; SMITH; SEARLE, 2003; SMITH, 2003).

3.2.1 Constitutive specification language

From the previously described elements, we introduce a language to specify the constitution of status functions. The constitutive specification, written based on the syntax given in Figure 9, defines the sets of status functions ($\mathcal{F}$) and constitutive rules ($\mathcal{C}$) of the institution. Each constitutive rule (*const_rule* in the grammar) has an identifier (*id*). Furthermore, the rules have the operator `count-as`, that performs the constitution of the status functions. The elements related to the context of the constitutive rule (t and m) are optional. The elements w in the grammar correspond to the w-formulae (syntax in the grammar 3.1) where `"not"`, `"false"`, and `"true"` correspond respectively to $\neg$, $\perp$, and $\top$. The element a of an event $(e,a) \in \mathcal{E}_{\mathcal{X}}$ is represented in the grammar by `sai__agent(`a`)`, that can be omitted to represent $a = \lambda$. Constants and atoms start by a lower case letter. Variables *var* are terms starting by an upper case letter.

$$const\_model ::= \mathcal{F}\ \ \mathcal{C}$$

Figura 9 – Grammar of the constitutive specification

Figure 10 shows the constitutive specification for the use case addressed in (BRITO et al., 2015b), where agents collaborate to manage crisis such as floodings, car crashes, etc. They act in an environment composed of geographic information systems (GIS) and of tangible tables (KUBICKI; LEPREUX; KOLSKI, 2012) where they put objects equipped with RFID tags on to signal their intended actions. The constitutive rules assign institutional meaning to the environment elements. For example, putting a *launch_object* on the coordinates (15,20) of a table signals the evacuation of the downtown (constitutive rule 3).

```
status_functions:
  agents: mayor, firefighter.
  events: evacuate(Zone).
  states: secure(Zone), insecure(Zone).
constitutive_rules:
                 /*** Agent-Status Functions constitutive rules ***/
   /*Actors carry the status functions according to their check in the tables*/
   1: Actor count-as mayor
            when checkin(table_mayor,Actor) while not(Other is mayor)|Other==Actor.
   2: Actor count-as firefighter
            when checkin(table_fire_brigade,Actor).

             /*** Event-Status Functions constitutive rules ***/
   /*Putting the ‘‘launch_object’’ on (15,20) means the evacuation of the downtown*/
   3: put_tangible(launch_object,15,20) count-as evacuate(downtown).
   /*Sending a message with the proper arguments means the evacuation of the downtown*/
   4: send_message(evacuation,downtown) count-as evacuate(downtown).

             /*** State-Status Functions constitutive rules ***/
   /*A zone in preventive phase is secure if it has at most 500 inhabitants*/
   5: security_phase(Zone,preventive) count-as secure(Zone)
             while nb_inhabit(Zone,X)& X<=500
   /*A zone in emergency phase is insecure*/
   6: security_phase(Zone,emergency) count-as insecure(Zone).
```

Figura 10 – Example of constitutive specification

3.3 CONSTITUTIVE INTERPRETATION

The previous section introduced the elements to specify how the institutional reality is constituted from the environment. This section, on its turn, defines how constitutive specifications are interpreted, animating thus the institutional reality of the system. Notice that this is a proposal of a possible approach to interpret constitutive specifications. Other approaches to interpret the same specifications, giving different semantics to the institutional reality, are possible.

In the following, Section 3.3.1 introduces the representations involved in the constitutive dynamics of SAI. These representations are important to describe, in sections 3.3.2 and 3.3.3, the two types of constitution in SAI, that are the first- and second-order constitution.

3.3.1 Preliminaries

The constitutive dynamics of SAI involves the actual state of the environment and the actual constitutive state (i.e. the standing SFAs), that are part of the whole SAI state. These states, as well as their components, are defined in this section.

Definition 3.3.1 (Environmental state) *The actual environmental state is represented by* $X = A_X \cup E_X \cup S_X$ *where (i)* A_X *is the set of agents participating in the system, (ii)* E_X *is the set of events occurring in the environment and (iii)* S_X *is the set of environmental properties describing the environmental state.*

Agents in A_X are represented by constants referring to their names. States in S_X are represented by atoms. Events in E_X are represented by pairs (e, a) where e is the event, identified by an atom, triggered by the agent a. Events can be triggered by actions of the agents (e.g. the utterance of a bid in an auction, the handling of an environmental artifact, etc) but can be also produced by the environment itself (e.g. a clock tick). In this case, events are represented by pairs (e, λ).

Definition 3.3.2 (Status function assignment) *Status function assignments are relations between environmental elements and status functions s.t. (i)* $A_F \subseteq A_X \times \mathcal{A}_{\mathcal{F}}$ *is the set of agent-status function assignments, (ii)* $E_F \subseteq E_X \times \mathcal{E}_{\mathcal{F}} \times A_X \cup \lambda$ *is the set of event-status function assignments and (iii)* $S_F \subseteq S_X \cup \{\lambda\} \times \mathcal{S}_{\mathcal{F}}$ *is the set of state-status function assignments.*

SFAs are established through the constitution of status functions (cf. sections 3.3.2 and 3.3.3). Elements of A_F are pairs $\langle a_X, a_{\mathcal{F}} \rangle$ meaning that the agent $a_X \in A_X$ has the status function $a_{\mathcal{F}} \in \mathcal{A}_{\mathcal{F}}$. As events are supposed to be considered at the individual agent level in normative systems (VOS; BALKE; SATOH, 2013), it is important to record the agent that causes an event-status function assignment. For this reason, elements of E_F are triples $\langle e_X, e_{\mathcal{F}}, a_X \rangle$ meaning that the event-status function $e_{\mathcal{F}} \in \mathcal{E}_{\mathcal{F}}$ is assigned to the event $e_X \in E_X$, either produced by the agent $a_X \in A_X$ or spontaneously produced (if $a_X = \lambda$). Elements of S_F are either pairs $\langle s_X, s_{\mathcal{F}} \rangle$ meaning that the state $s_X \in S_X$ carries the status function $s_{\mathcal{F}} \in \mathcal{S}_{\mathcal{F}}$ or pairs $\langle \lambda, s_{\mathcal{F}} \rangle$ representing the freestanding assignment of the state-status function $s_{\mathcal{F}}$. By this definition it is clear that while agent- and event-status function have counterparts in the environment, state-status functions admit freestanding assignment. This decision is taken because it is considered that (i) there might have both environment and institutional conditions – not necessarily involving environmental states – that mean states holding in the institutional reality but (ii) it does not make sense to have some institutional representation of agents and events without corresponding agents and events actually existing in the environment.

Definition 3.3.3 (Constitutive state) *The constitutive state of a SAI is the set of the existing SFAs. It is represented by* $F = A_F \cup E_F \cup S_F$.

To check the introduced environmental and constitutive states, we define w-formulae $w \in W$ following the BNF grammar rule (3.1):

$$w ::= x|\neg w|w \vee w|w \wedge w|x \textbf{ is } y|\bot|\top \tag{3.1}$$

s.t. x and y are terms of the first-order logic language.

Considering a model $M = \langle F, X, \mathcal{F} \rangle$, the semantics of w-formulae is defined as follows[6]:

$$\begin{aligned} M \models x \text{ iff } \exists \theta :& (x\theta \in E_X \vee x\theta \in S_X) \vee & (3.2)\\ & (\exists e_X : e_X \textbf{ is } x\theta) \vee \\ & (\exists s_X : s_X \textbf{ is } x\theta) \\ M \models x \textbf{ is } y \text{ iff } \exists \theta :& (x\theta \in A_X \wedge y\theta \in \mathcal{A}_{\mathcal{F}} \wedge \langle x\theta, y\theta \rangle \in A_F) \vee & (3.3)\\ & (x\theta \in E_X \wedge x = (e,a) \wedge y\theta \in \mathcal{E}_{\mathcal{F}} \wedge \langle e\theta, y\theta, a\theta \rangle \in E_F) \vee \\ & (x\theta \in S_X \wedge y\theta \in \mathcal{S}_{\mathcal{F}} \wedge \langle x\theta, y\theta \rangle \in S_F) \end{aligned}$$

Definition 3.3.4 (SAI state) *The SAI state is composed of an environmental state X, a constitutive state F, and a normative state N. It is represented by* $SAI_{Dyn} = \langle X, F, N \rangle$.

The formal representations of X and F are introduced respectively in definitions 3.3.1 and 3.3.3. The normative state N depends on the dynamics of each normative model possibly composing the institution. Chapter 4 details how norms take part to SAI. Here, N subsumes the state of all the norms taking part in the institution.

Definition 3.3.5 (SAI history) *The history of a SAI is the sequence of its* $i \in \mathbb{N}$ *states (where* $\mathbb{N}$ *is the set of the natural numbers).*

The SAI state at the i^{th} step of its history is represented by $SAI^i_{Dyn} = \langle X^i, F^i, N^i \rangle$ where $X^i = A^i_X \cup E^i_X \cup S^i_X$ and $F^i = A^i_F \cup E^i_F \cup S^i_F$. A sequence of steps starting from the step s finishing in the step z (s.t. $s \in \mathbb{N}$ and $z \in \mathbb{N}$) is noted as $SAI^s_{Dyn} \cdots SAI^z_{Dyn}$. Sequences of environmental, constitutive, and normative states are similarly noted, respectively, as $X^s \cdots X^z$, $F^s \cdots F^z$, and $N^s \cdots N^z$.

---

[6] In this thesis, a *substitution* is always represented by θ. A *substitution* is a finite set of pairs $\{\alpha_1/\beta_1, \cdots \alpha_n/\beta_n\}$ where α_i is a variable and β_i is a term. If θ is a substitution and ρ is a literal, then $\rho\theta$ is the literal resulting from the replacement of each α_i in ρ by the corresponding β_i (BRACHMAN; LEVESQUE, 2004).

The SAI history evolves from a step i to $i+1$ due to changes in the environment, that may trigger changes in the constitutive state that, on its turn, may trigger changes in the normative state. We leave the normative dynamics aside for now, focusing on the constitutive dynamics, that is based on the environmental one. Figure 11 illustrates this dynamics along three steps of a SAI history. In the first step, an environmental element is added to X reflecting some change in the real environment. This change in the environmental representation X produces new SFAs, leading to changes in the constitutive state F. The environment then goes to a new state (two), that does not constitute any condition neither to create new SFAs nor to revoke the existing ones. The environment then changes again, leading SAI to the state three, where changes are also produced in the constitutive state.

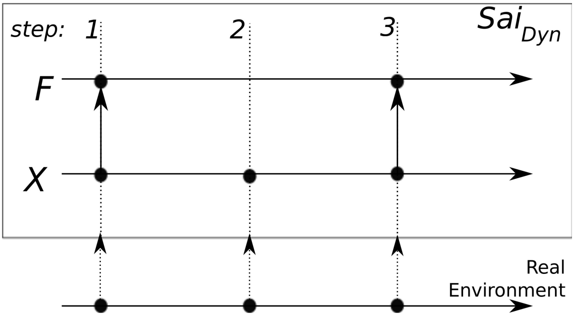


Figura 11 – SAI dynamics overview

Constitutive rules can specify two kinds of constitution of status functions to produce SFAs: *first-order constitution* and *second-order constitution*. Sections 3.3.2 and 3.3.3 explain how SFA due to first- and second-order constitution can be deduced from a given SAI history.

3.3.2 First-order constitution

Constitutive rules specifying *first-order constitution* explicitly define that agent-, event-, and state- status functions are assigned to agents, events, and states from the environment. In the following, from definitions 3.3.6 to 3.3.8 define how this kind of constitution is produced. To formally define constitution, we use functions to define the set of SFAs that can be deduced from given environmental and constitutive states.

Definition 3.3.6 (First-order constitution of agent-status-functions) *Given a set $\mathcal{F}$ of status functions, a set $\mathcal{C}$ of constitutive rules, and histories X and F of z environmental and constitutive states, the set of agent-status function assignments due to first-order constitution that can be deduced in the i^{th} step ($0 \leq i \leq z$) is given by the function $f\text{-}const_a$ defined as follows:*

$$\begin{aligned} f\text{-}const_a(\mathcal{F},\mathcal{C},X^0\cdots X^z,F^0\cdots F^z,i) = \{\langle x,y\rangle | & x \in A^i_X \wedge y \in \mathcal{A}_{\mathcal{F}} \wedge \\ & \exists c \in \mathcal{C}\ \exists \theta\ \exists s \in \mathbb{N}\ \forall k \in [s,i] : \\ & (E^s_X \cup E^s_F \models t\theta) \wedge \\ & (X^k \cup F^k \models m\theta) \wedge \\ & x'\theta = x\} \end{aligned} \tag{3.4}$$

$$s.t.\ c = \langle x',y,t,m\rangle$$

Informally, the function $f\text{-}const_a$ defines that (i) if exists a constitutive rule $\langle x',y,t,m\rangle$ whose element t, under a substitution θ, represents an event occurred at the step s and (ii) if along all the steps k from s to i the formula m, under θ, is entailed by the environmental and constitutive states, then the agent identified by the element x' under θ carries the agent-status function y in the step i. Note that the function returns SFA only for the agents that are participating in the system. If an agent a is participating in the system in the step i but leaves the system in the step $i+1$ (i.e. $a \in A^i_X$ and $a \notin A^{i+1}_X$), then the SFAs of the a returned by the function $f\text{-}const_a$ in the step i are not returned in the step $i+1$. The management of the constitutive state, when based on this function, can drop the SFA of agents that have leaved the system (cf. Section 3.3.4).

The function $f\text{-}const_a$ also explicits our proposed approach to deal with combined instantaneous events and fluent states as conditions to constitution when it defines that a SFA belongs to the constitutive state if m holds in all steps k from the occurrence of t (at the step s) until the step i. Some points to observe in this definition are: (i) the repetition of the event t does not affect the SFA and (ii) a SFA is dropped if m ceases to hold and is not undropped if the m turns to hold (unless the event t happens again while m is again holding).

The rule 1 in the Figure 10 defines a first-order constitution of an agent-status function. If $checkin(table\_mayor, bob) \in E^1_X$, meaning that the agent bob has checked in the $table\_mayor$ at the step 1, then bob carries the status function $mayor$ (i.e. $\langle bob, mayor\rangle \in f\text{-}const_a(\mathcal{F},\mathcal{C},X^0\cdots X^z,F^0\cdots F^z,i)$) for all steps i, starting from the 1^{st} one, while bob participates in the system (considering $\theta = \{Agent/bob\}$).

Definition 3.3.7 (First-order constitution of state-status-functions)
Given a set $\mathcal{F}$ of status functions, a set $\mathcal{C}$ of constitutive rules, and histories X and F of z environmental and constitutive states, the set of state-status-function assignments due to first-order constitution that can be deduced in the i^{th} step ($0 \leq i \leq z$) is given by the function $f\text{-}const_s$ defined as follows:

$$
\begin{aligned}
f\text{-}const_s(\mathcal{F},\mathcal{C},X^0\cdots X^z,F^0\cdots F^z,i) = \{\langle x,y\rangle | & ((x=\lambda)\vee(x\in S_X^i))\wedge \\
& (y\in\mathcal{S}_{\mathcal{F}})\wedge \\
& \exists c\in\mathcal{C}\ \exists\theta\ \exists s\in\mathbb{N}\ \forall k\in[s,i]: \\
& (E_X^s\cup E_F^s\models t\theta)\wedge \\
& (X^k\cup F^k\models m\theta)\wedge \\
& x'\theta = x\}
\end{aligned}
\tag{3.5}
$$

$$s.t.\ c=\langle x',y,t,m\rangle$$

Similar to the constitution of agent-status functions, (i) a SFA is only assigned to a state $x \in S_X$ if x actually holds in the environment and (ii) the constitution of state-status functions is conditioned by the holding of m in all steps from the occurrence of the event t. Furthermore, the function $f\text{-}const_s$ explicits our conception that the constitution of state-status functions may result in freestanding assignments.

The rule 6 in the Figure 10 defines a first-order constitution of a state-status function. If $security\_phase(downtown, emergency) \in S_X^3$, meaning that the GIS points the downtown as being in emergency phase of a crisis, then the assignment $\langle security\_phase(downtown, emergency), insecure(downtown)\rangle$ can be deduced from the step 3 while the GIS remains indicating $security\_phase(downtown, emergency)$.

Definition 3.3.8 (First-order constitution of event-status-functions)
Given a set $\mathcal{F}$ of status functions, a set $\mathcal{C}$ of constitutive rules, and histories X and F of z environmental and constitutive states, the set of event-status-function assignments due to first-order constitution that can be deduced in

the i^{th} step ($0 \leq i \leq z$) is given by the function $f\text{-}const_e$ defined as follows:

$$\begin{aligned} f\text{-}const_e(\mathcal{F},\mathcal{C},X^0\cdots X^z,F^0\cdots F^z,i) = \{\langle e,y,a\rangle | &(e,a) \in E_X^i \wedge y \in \mathcal{E}_{\mathcal{F}} \wedge \\ &\exists c \in \mathcal{C}\ \exists \theta : \\ &(E_X^i \cup E_F^i \models t\theta) \wedge \\ &(X^i \cup F^i \models m\theta) \wedge \\ &x' = (e',a') \wedge \\ &e'\theta = e \wedge a'\theta = a\} \end{aligned} \quad (3.6)$$

$$s.t.\ c = \langle x',y,t,m\rangle$$

Compared to agent- and state-status functions, the constitution of event-status functions is differently related to the SAI history. Event-status function assignments are assumed to hold only in the step in which the events $x' \in E_X$ and $t \in E_X$ happen, mimicking, thus, in the constitutive level, the atomic nature of the environmental events (CASSANDRAS; LAFORTUNE, 2006). Thus, the holding of m during many steps of the SAI history does not imply in the holding of an event-status function assignment.

The rule 3 in the Figure 10 defines the first-order constitution of an event-status function. If $(put\_tangible(launch\_object,15,20),tom) \in E_X^2$ meaning that tom has has put the a $launch\_object$ on the coordinates (15,20) of the table at the step 2, then the assignment $\langle put\_tangible(launch\_object,15,20),evacuate(downtown),tom\rangle$ holds in the step 2, i.e.:

$$\begin{aligned} &\langle put\_tangible(launch\_object,15,20),evacuate(downtown),tom\rangle \in \\ &\qquad f\text{-}const_e(\mathcal{F},\mathcal{C},X^0\cdots X^z,F^0\cdots F^z,2) \end{aligned}$$

3.3.3 Second-order constitution

Constitutive rules specifying second-order constitution define that a status function counts as another status function. But even specifying a relation between two status functions, the assignments resulting of the second-order constitution are also relations between status functions and environmental elements. That is to say, whenever a status function s_1 counts as a status function s_2, all the elements constituting s_1 constitute also s_2. Consider, for example, that the constitutive rule `firefighter` **count-as** `security_expert` is added to the specification in Figure 10 (s.t.

$\{firefighter, security\_expert\} \subset \mathcal{A}_{\mathcal{F}}$). In this case, the agent-status function *security_expert* is actually assigned to every agent carrying the status function *firefighter*.

Definition 3.3.9 (Second-order constitution of agent-status-functions) *Given a set $\mathcal{F}$ of status functions, a set $\mathcal{C}$ of constitutive rules, and histories X and F of z environmental and constitutive states, the set of agent-status function assignments due to second-order constitution that can be deduced in the i^{th} step ($0 \leq i \leq z$) is given by the function s-const$_a$ below:*

$$\begin{aligned} s\text{-}const_a(\mathcal{F}, \mathcal{C}, X^0 \cdots X^z, F^0 \cdots F^z, i) = \{\langle x, y\rangle | x \in A^i_X \wedge y \in \mathcal{A}_{\mathcal{F}} \wedge \\ \exists c \in \mathcal{C}\ \exists \theta\ \exists s \in \mathbb{N}\ \forall k \in [s, i] : \\ (E^s_X \cup E^s_F \models t\theta) \wedge \\ (X^k \cup F^k \models m\theta) \wedge \\ x'\theta \in \mathcal{A}_{\mathcal{F}} \wedge \langle x, x'\theta\rangle \in A^i_F\} \end{aligned} \tag{3.7}$$

$$s.t.\ c = \langle x', y, t, m\rangle$$

Informally, if there is a constitutive rule $\langle x', y, t, m\rangle$ whose element x', under a substitution θ, corresponds to a status function already assigned to an agent by x, then this agent carries also the status function $y \in \mathcal{A}_{\mathcal{F}}$ (subject to the conditions t and m, as in the first-order constitution (Definition 3.3.6)). When the agent x ceases to carry the status function $x'\theta$, it also ceases to carry the status function y.

In the crisis scenario, we can imagine the agent-status function *authority* and the constitutive rule `mayor` **count-as** `authority`. If *bob* **is** *mayor* at the i^{th} step (i.e. $\langle bob, mayor\rangle \in A^i_F$), then $\langle bob, authority\rangle \in s\text{-}const_a(\mathcal{F}, \mathcal{C}, X^0 \cdots X^z, F^0 \cdots F^z, i)$ (for $i \leq z$). Informally, the rule states that an agent having the status function of *mayor* counts as an *authority* and, as *bob* has the status function of *mayor*, he has also the status function of *authority*.

Definition 3.3.10 (Second-order constitution of state-status-functions) *Given a set $\mathcal{F}$ of status functions, a set $\mathcal{C}$ of constitutive rules, and histories X and F of z environmental and constitutive states, the set of state-status function assignments due to second-order constitution that can be deduced* *in the i^{th} step ($0 \leq i \leq z$) is given by the function s-const$_s$ below:*

$$
\begin{aligned}
\textit{s-const}_s(\mathcal{F},\mathcal{C},X^0\cdots X^z,F^0\cdots F^z,i) = \{\langle x,y\rangle \mid\; & (x \in S^i_X \vee x = \lambda) \wedge y \in \mathcal{S_F} \wedge \\
& \exists c \in \mathcal{C}\ \exists \theta\ \exists s \in \mathbb{N}\ \forall k \in [s,i] : \\
& (E^s_X \cup E^s_F \models t\theta) \wedge \\
& (X^k \cup F^k \models m\theta) \wedge \\
& x'\theta \in \mathcal{S_F} \wedge \langle x, x'\theta\rangle \in S^i_F\}
\end{aligned}
\tag{3.8}
$$

$$s.t.\ c = \langle x', y, t, m\rangle$$

If there is a constitutive rule $\langle x', y, t, m\rangle$ whose element x, under a substitution θ, corresponds to a status function already assigned to a state x, then this state carries also the status function $y \in \mathcal{S_F}$ (subject to the conditions t and m, as in the first-order constitution (Definition 3.3.7)). When x ceases to carry the status function $x'\theta$, it also ceases to carry the status function y.

We can consider the state-status function *red_alert* in the crisis scenario and the constitutive rule `insecure(Zone)` **count-as** `red_alert`. If $\langle security\_phase(downtown, emergency), insecure(downtown)\rangle \in S^i_F$, then $\langle security\_phase(downtown, emergency), red\_alert\rangle \in$ *s-const*$_s(\mathcal{F},\mathcal{C},X^0\cdots X^z,F^0,\cdots F^z,i)$ (for $i \leq z$).

Definition 3.3.11 (Second-order constitution of event-status-functions) *Given a set $\mathcal{F}$ of status functions, a set $\mathcal{C}$ of constitutive rules, and histories X and F of z environmental and constitutive states, the set of event-status function assignments due to second-order constitution that can be deduced in the i^{th} step ($0 \leq i \leq z$) is given by the function s-const$_e$ below:*

$$
\begin{aligned}
\textit{s-const}_e(\mathcal{F},\mathcal{C},X^0\cdots X^z,F^0\cdots F^z,i) = \{\langle e,y,a\rangle \mid\; & (e,a) \in E^i_X \wedge y \in \mathcal{E_F} \wedge \\
& \exists c \in \mathcal{C}\ \exists \theta\ : \\
& (E^i_X \cup E^i_F \models t\theta) \wedge \\
& (X^i \cup F^i \models m\theta) \wedge \\
& x'\theta \in \mathcal{E_F} \wedge \langle e, x'\theta, a\rangle \in E^i_F\}
\end{aligned}
\tag{3.9}
$$

$$s.t.\ c = \langle x', y, t, m\rangle$$

If there is a constitutive rule $\langle x', y, t, m \rangle$ whose element x', under a substitution θ, corresponds to a status function already assigned to the event $(e, a) \in E_X$, then this event also carries the status function $y \in \mathcal{E}_{\mathcal{F}}$ (subject to the conditions t and m, as in the first-order constitution (Definition 3.3.8)). The assignment of y to (e, a) holds while the assignment of x' to (e, a) holds.

Consider, for example, that the specification of Figure 10 is enriched with the event-status function *security_procedure* and with the constitutive rule `evacuate(Zone)` **count-as** `security_procedure`. If the agent *tom* has put a *launch_object* on the coordinates (15,20) of the table at the step i, then by the rule 3, we have that $put\_tangible(launch\_object, 15, 20), evacuate(downtown), tom) \in E^i_F$ and, by the introduced rule, we have that $\langle put\_tangible(launch\_object, 15, 20), security\_procedure, tom \rangle \in s\text{-}const_e(\mathcal{F}, \mathcal{C}, X^0 \cdots X^z, F^0 \cdots F^z, i)$ because (i) the term x of the rule is an event-status-function that (ii) is already assigned to the event $put\_tangible(launch\_object, 15, 20)$.

From the definitions 3.3.9 to 3.3.11 we can see how specifying status functions to be assigned to other status functions allows to ground the institution in the environment while it enables different kinds of manipulations inside the constitutive level, such as the definition of multiple levels of abstraction (defining, for example, that the status functions y_1 counts as y_2, that, on its turn, counts as y_3), as well allowing to define relations inside the constitutive level such as generalization (e.g. y_1 and y_2 count as y_3), etc.

3.3.4 Building the SAI Constitutive State

From the previous definitions, it is possible to define how the constitutive state of an institution is built. First, for an institution where the set of status functions is $\mathcal{F}$ and the set of constitutive rules is $\mathcal{C}$, we define $const(X^0 \cdots X^i, F^0 \cdots F^i)$ as the set of all the SFAs that can be deduced given environmental and constitutive histories of size i:

$$\begin{aligned} const(X^0 \cdots X^i, F^0 \cdots F^i) = \langle & \textit{f-const}_a(\mathcal{F}, \mathcal{C}, X^0 \cdots X^i, F^0 \cdots F^i, i) \cup \\ & \textit{s-const}_a(\mathcal{F}, \mathcal{C}, X^0 \cdots X^i, F^0 \cdots F^i, i), \\ & \textit{f-const}_e(\mathcal{F}, \mathcal{C}, X^0 \cdots X^i, F^0 \cdots F^i, i) \cup \\ & \textit{s-const}_e(\mathcal{F}, \mathcal{C}, X^0 \cdots X^i, F^0 \cdots F^i, i), \\ & \textit{f-const}_s(\mathcal{F}, \mathcal{C}, X^0 \cdots X^i, F^0 \cdots F^i, i) \cup \\ & \textit{s-const}_s(\mathcal{F}, \mathcal{C}, X^0 \cdots X^i, F^0 \cdots F^i, i) \rangle \end{aligned} \tag{3.10}$$

Additions and revocations of SFAs may create conditions for new constitutions. A constitutive state F^i of a SAI history is *closed under constitution* if all the SFAs that can be deduced from the step $i-1$ are in F^i. The closure of SFAs under environmental and constitutive histories of size i is given by the function F^* as follows.

$$F^*(X^0 \cdots X^i, F^0 \cdots F^i) = \begin{cases} F^i \text{ if } F^i = F^{i'} \\ \\ F^*(X^0 \cdots X^i, F^0 \cdots F^{i'}) \text{ otherwise} \end{cases} \tag{3.11}$$
$$\text{s.t. } F^{i'} = const(X^0 \cdots X^i, F^0 \cdots F^i)$$

Informally, the set of SFA deductible from F^i is computed until there is not any SFA to add or to remove, when the closure of F is finally found. New constitutive states are built based on the constitutive closure of a previous state. Handling the constitutive state is up to some constitutive monitor implementing the transition rule 3.12. Notice that the environmental and normative states X and N do not change.

$$\frac{SAI^i_{Dyn} = \langle X^i, F^i, N^i \rangle \quad F^i \neq const(X^0 \cdots X^i, F^0 \cdots F^i)}{\langle X^i, F^i, N^i \rangle \rightarrow \langle X^i, F^*(X^0 \cdots X^i, F^0 \cdots F^i), N^i \rangle} \tag{3.12}$$

Informally, if SAI is in a state i s.t. a new constitutive state can be deduced from the current one, then the new constitutive state is the closure of the previous one.

3.4 EXAMPLE OF CONSTITUTIVE DYNAMICS

To illustrate the constitutive dynamics, we consider the scenario introduced in Section 3.2.1 and the constitutive specification illustrated in Figure 10. We consider five steps of the environmental dynamics. In each step, the environmental state changes causing changes in the constitutive state. This dynamics is described below and summarized in the Table 2:

- **Step 1.** GIS indicate that the properties *security_phase(downtown,preventive)* and *(nb_inhabit(downtown,200)* hold in the environment, meaning that (i) the downtown is on preventive phase of the crisis management and (ii) the downtown has 200 inhabitants. By the constitutive rule 5, the institution considers the downtown as a secure zone. At this moment, the actor *bob* checks in the *table_mayor* and the actors *tom*, *jim*, and *ana* check in the *table_fire_brigade*. By the constitutive rules 1 and 2, *bob* is considered by the institution as the *mayor* while *tom*, *jim* and *ana* are considered *firefighter*.

- **Step 2**. *Bob* puts the *launch_object* on the coordinates (15,20). By the constitutive rule 3, this means, from the institutional perspective, the evacuation of the downtown.

- **Step 3**. After the evacuation performed by *bob*, for some reason, the downtown has 50 inhabitants. The security phase of the crisis changes from preventive to emergency, and, from the institutional perspective, the downtown is insecure (constitutive rule 6).

- **Step 4**. *Tom* puts the *launch_object* on the coordinates (15,20) of the table while *jim* sends a message. Both the actions count as the evacuation of the downtown (constitutive rules 3 and 4).

- **Step 5**. The security phase of the crisis becomes again preventive, and, from the institutional perspective, the downtown is again secure (constitutive rule 5).

3.5 CONCLUSIONS

In Searle's theory, the building blocks of the institutional reality are the elements constituted from the environment according to the specified constitutive rules. Inspired by such idea, the first question posed at the beginning of the chapter is answered by defining the institutional reality in an artificial institution as composed of *status functions assignments*, i.e. status functions constituted according to *constitutive rules*. This chapter also introduces the

Tabela 2 – Evolution of environmental and constitutive states

Step	Environmental State (X)	Constitutive State (F)
1	$A_X = \{bob, tom, jim, ana\}$ $E_X = \{(checkin(table\_maior), bob),$ $(checkin(table\_fire\_brigade), tom),$ $(checkin(table\_fire\_brigade), jim),$ $(checkin(table\_fire\_brigade), ana)\}$ $S_X = \{security\_phase(downtown, preventive),$ $nb\_inhabit(downtown, 200)\}$	$A_F = \{\langle bob, mayor\rangle, \langle tom, firefighter, \rangle,$ $\langle jim, firefighter\rangle, \langle ana, firefighter\rangle\}$ $S_F = \{\langle security\_phase(downtown, preventive),$ $secure(downtown)\rangle\}$
2	$A_X = \{bob, tom, jim, ana\}$ $E_X = \{(putTangible(launch\_object, 15, 20), bob)\}$ $S_X = \{security\_phase(downtown, preventive),$ $nb\_inhabit(downtown, 200)\}$	$A_F = \{\langle bob, mayor\rangle, \langle tom, firefighter, \rangle,$ $\langle jim, firefighter\rangle, \langle ana, firefighter\rangle\}$ $E_F = \{\langle putTangible(launch\_object, 15, 20),$ $evacuate(downtown), bob\rangle\}$ $S_F = \{\langle security\_phase(downtown, preventive),$ $secure(downtown)\rangle\}$
3	$A_X = \{bob, tom, jim, ana\}$ $S_X = \{security\_phase(downtown, emergency),$ $nb\_inhabit(downtown, 50)\}$	$A_F = \{\langle bob, mayor\rangle, \langle tom, firefighter, \rangle,$ $\langle jim, firefighter\rangle, \langle ana, firefighter\rangle\}$ $S_F = \{\langle security\_phase(downtown, emergency),$ $insecure(downtown)\rangle\}$
4	$A_X = \{bob, tom, jim, ana\}$ $E_X = \{(putTangible(launch\_object, 15, 20), tom),$ $(send\_message(evacuation, downtown), jim)\}$ $S_X = \{security\_phase(downtown, emergency),$ $nb\_inhabit(downtown, 50)\}$	$A_F = \{\langle bob, mayor\rangle, \langle tom, firefighter, \rangle,$ $\langle jim, firefighter\rangle, \langle ana, firefighter\rangle\}$ $E_F = \{\langle putTangible(launch\_object, 15, 20),$ $evacuate(downtown), tom\rangle,$ $\langle send\_message(evacuation, downtown),$ $(evacuate(downtown), jim)\rangle\}$ $S_F = \{\langle security\_phase(downtown, emergency),$ $insecure(downtown)\rangle\}$
5	$A_X = \{bob, tom, jim, ana\}$ $S_X = \{security\_phase(downtown, preventive),$ $nbInhabit(downtown, 50)\}$	$A_F = \{\langle bob, mayor\rangle, \langle tom, firefighter, \rangle,$ $\langle jim, firefighter\rangle, \langle ana, firefighter\rangle\}$ $S_F = \{\langle security\_phase(downtown, preventive),$ $secure(downtown)\rangle\}$

elements necessary to specify and to produce the assignments of status functions, building the *constitutive state*, that is the proposed representation of the institutional reality.

It is assumed that the environmental elements referred by the regulation, that should have a counterpart in the institutional reality, are agents, events, and states. The different nature of these elements must be explicit in the institutional reality for three main reasons. First, to enable consistent relations between components of the institutional reality and their environmental counterparts, making clear, for example, that the status function *firefighter* is assignable to an agent and *evacuate* is assignable to an event. Second, to enable consistent relations between the elements of the institutional reality and the components of the norms, making clear, for example, that *firefighter* corresponds to what norms deem as *attribute* while *evacuate* may be the *aim* or some *condition* of a norm.[7] Finally, to manage the constitution of the institutional elements taking into account the dynamics related to the natures of their environmental counterparts. These reasons answer the second question posed at the beginning of this chapter: the elements of the institutional reality must be distinguishable according to the nature of their environmental counterpart to consistently ground the institutional reality in the environment and also to consistently couple the norms with the institutional reality. For these reasons, SAI considers explicitly the three kinds of status functions.

Having defined these three kinds of status functions, it is possible to answer the third question posed at the beginning of this chapter. Agent-, event-, and state-status functions are elements of different natures within the institution. The different natures of these elements is reflected, first, in the representations of the status function assignments. Agent-status function assignments are relations between agents acting in the environment and agent-status functions. State-status functions may be either relations between state-status functions and environmental states or freestanding assignments, where there is not an environmental counterpart for the state-status function. Event-status functions are relations between event-status functions, environmental events, and agents acting in the environment. This ternary relation captures event-status function assignments in individual agent level, i.e. event-status function assignments are explicitly attributed to a triggering agent.

The different natures of the status functions are also considered in the management of the constitutive state, as described in Section 3.3. We defined a uniform constitutive dynamics considering that SFA may have specific life cycles according to their nature. To achieve it, we first defined the life cycles of the SFAs that even being produced by similar definition of constitutive rules, may be distinguished into: (i) agent-status function assignments hold-

---

[7] This point is addressed in details in Chapter 4.

ing only while the agent that carries the status function participates to the system, (ii) state-status function assignments holding while the state carrying the status function holds in the environment and (iii) event-status function assignments holding only during a single step of the SAI history. These definitions have been then complemented by the explicitation of the instantaneous and fluent expressions conditioning these constitutions. We captured important properties on this dynamics such as: proper dynamics of status function assignment for event, state or agents, stability of constituted status functions wrt. repetition of events, dropping of constituted status function as soon as state condition is no more holding, etc.

With the model of the institutional reality presented in this chapter it is possible to design and animate the institutional reality of the systems as the result of the environmental dynamics. The next chapter analyses how the regulation in MAS can be based on the institutional reality as conceived here.

4 REGULATION BASED ON THE INSTITUTIONAL REALITY

SAI considers that, in an institution, the regulation provided by the norms is based on the institutional reality, that is the *institutional* interpretation of the environment where the agents act. Such institutional reality is represented, in SAI, by the constitutive state, defined in Chapter 3. It is not assumed, however, a specific model of norms to regulate the MAS on top of the constitutive state. To the contrary, it is assumed that norms in general, following different models, expressed in different ways and having different dynamics, may be part of the institution, providing the regulation based on the constitutive state.

The different normative models proposed in the literature are not explicitly conceived to be based on a constitutive state. Having, thus, in one hand, a well defined representation of institutional reality and, in other hand, many ways to represent the regulation, this chapter aims to achieve another objective of this thesis, that is to define how the regulation, as proposed in the literature, couples with the proposed representation of institutional reality. Part of this content is also presented in (BRITO; HÜBNER; BOISSIER, 2015b). The questions to be answered in this chapter are:

1. Can the norms, as currently conceived in the literature, be based on a unified representation of institutional reality?
2. How to base norms following different models on the same institutional reality?

To answer the first question, we depart from the general notion of norms provided by the Section 2.2.1 to have a general view of how norms, as a whole, can be related with the SAI constitutive state (Section 4.1). Answering the second question requires to go in the details of the normative models. For this reason, we analyse in Section 4.2 how two normative models proposed in the literature can be coupled with the SAI constitutive state.

4.1 REPRESENTING NORMS THROUGH STATUS FUNCTIONS

Since it is not possible to check particularly each existing normative model, it is not possible to answer the first posed question by showing that all of the existing normative models can be based on the SAI constitutive state. We adopt thus a different path to answer that question. Following the exposed in Section 2.2.1.1, we consider that norms, in general, are composed of the ADICO elements. To analyse, without considering each existing normative model, whether norms as conceived in literature can be based on the unified representation of institutional reality conceived by SAI, we analyse below how the ADICO elements can be related to the constitutive state.

- The ***Attribute*** of a norm, when based on the constitutive state, is an agent-status function because norms, when being part of an institution as conceived by SAI, specify the expected behaviour of agents that carry a function instead of to specify the expected behaviour of the concrete agents acting in the environment.

- The ***aIm*** is either an event- or a state-status function because, from the institutional perspective, all that the agents can do to behave as expected by the institution is to produce (or avoiding to produce), in the environment, events and states that carry event- and state-status functions.

- The ***Conditions*** refer to the whole constitutive state as, in SAI, the whole constitutive state can define the circumstances under which the norms must (or do not need to) be followed. Conditions over the whole constitutive state are expressed through sf-formulae $w_{\mathcal{F}} \in W_{\mathcal{F}}$, that are a subclass of the w-formulae (cf. grammar 3.1) whose atoms are either event- and state-status functions or expressions of the type x **is** y. It is assumed that conditions can also range over the very normative state (e.g. a norm can be activated when another one is violated). In this case, we assume these conditions as expressed through of n-formulae $w_N \in W_N$, whose syntax depends on the specific normative models.[1]

- The ***Or-else*** element of the norms have two possible approaches identified in Section 2.2.1.1: *institutional feedback* and *norm chaining*. Institutional feedback does not fits in the SAI approach because, for imposing some change in the environment, norms must refer to environmental elements, undermining the SAI premise of norms based exclusively on status functions. An alternative to use institutional feedback in SAI would be to define the or-else element in terms of status functions so that the normative platform should induce environmental dynamics that produce the corresponding constitutive state. But in this case, again, the normative platform should contain some model of the environment and the regulation would not be fully based on the institutional reality. Norm chaining seems to be a more suitable approach when regulation is inserted in the SAI constitutive state because, instead of specifying

---

[1] An example of syntax of n-formula is given in the grammar 4.5.

> what is expected from the normative platform, it specifies what is expected from the agents as the reaction to norm violations. Such expectations can be expressed in terms of status functions. Since the violation conditions are expressed through status functions, these same conditions can express activation conditions of other norms, that express how agents should behave in the case of such violation conditions. Depending on the normative model, norm chaining can be also an internal concern of the normative platform: if the normative language and platform provides means to check the state of the norms as violated, such violated state could be part of the activation condition of other norm without referring to status functions.

This analysis answers the first posed question at the beginning of this chapter, showing that (and how) norms, in general, can be *expressed* in terms of the elements that compose the constitutive state. Analysing how norms are *managed* when they are based on the constitutive state requires to consider particularly the different normative models, as they may have different conceptions for the lifecycle of the norms instances. The management of norms based on the constitutive state is addressed in the next section, that details how specific normative models can be based on the institutional reality as conceived by SAI.

4.2 COUPLING NORMATIVE MODELS WITH SAI

The previous section discussed how the ADICO elements, in terms of which norms are expressed, are related to the SAI constitutive state. As described in Section 2.2.1, the different existing normative models consider the ADICO elements in different ways to represent norms and to manage the regulation. This section checks with more details how different normative models, with their particular representations and management of norms, can base their regulation on the institutional reality represented by the SAI constitutive state.

Normative models, usually conceived without considering SAI, look to the "state of the world" to check the agents' expected behaviour. When norms are part of SAI, the "state of the world" that norms must look to is the constitutive state. But the elements to be actually regulated are the environmental ones. Such duality – norms being directed to status functions but regulating the environmental elements – raises challenges related to conceive how the environmental elements of different natures, abstracted under the notion of constitution, are taken into account in the norm lifecycle. For example, considering the norm *"a bidder is obliged to bid"*, it is necessary to define (i) how to monitor the norm taking into account all the agents considered as bidders, (ii) how to proceed when obliged agents are no longer considered as bidders or (iii) how to verify its compliance when many actions are considered as a bid (is the norm compliance conditioned to the performance of all of these actions or of at least one of them?).

Addressing such challenges requires to define (i) how the "world" represented by the constitutive elements is captured by the representations of norms and norm instances and (ii) how the different components of the norms are evaluated considering the different nature of the constituted elements in the different states of the lifecycle of the norm instances. In the following, we address the coupling of two normative models proposed in the literature with the SAI constitutive state. For each one of them, we (i) briefly describe the model, (ii) semantically align the normative representations – that are possibly different for each normative model – and the constitutive ones, and (iii) define how the dynamics of the social order that norms aim to achieve – that is also possibly different for each normative model – is animated by the dynamics of the constitutive state described in Section 3.3.

4.2.1 Coupling the norms of (PANAGIOTIDI; ÁLVAREZ-NAPAGAO; VÁZQUEZ-SALCEDA, 2013).

This section describes how the normative model proposed in (PANAGIOTIDI; ÁLVAREZ-NAPAGAO; VÁZQUEZ-SALCEDA, 2013) can base its regulation on the SAI constitutive state. Section 4.2.1.1 describes the normative model, Section 4.2.1.2 describes how the particular representations of the normative model are aligned with the constitutive representations of SAI, and Section 4.2.1.3 defines how the particular dynamics of the normative model is aligned is animated based on the constitutive dynamics.

4.2.1.1 The normative model

This section briefly describes the normative model of (PANAGIOTIDI; ÁLVAREZ-NAPAGAO; VÁZQUEZ-SALCEDA, 2013), firstly defining the norms that compose a normative specification and then defining norm instances (i.e. norms enacted in the real world) and their dynamics. The focus is on the elements that are essential to couple the norms on the constitutive state. More details about the normative model can be found in (PANAGIOTIDI; ÁLVAREZ-NAPAGAO; VÁZQUEZ-SALCEDA, 2013).

Definition 4.2.1 (Norm) *A norm n is a tuple* $n = \langle \alpha, c_a, c_m, c_d, c_r, c_t \rangle$ *where (i)* α *is the agent obliged to comply with the norm, (ii)* c_a *is the activation condition of the norm, (iii)* c_m *is the maintenance condition, (iv)* c_d *is the deactivation condition, (v)* c_r *is the repair condition, and (vi)* c_t *is the timeout condition. The set of all norms, noted* $\mathcal{N}$*, is called a normative specification.*

The element α is an agent identifier and the remainder c elements are expressed in first order predicate language with connectives $\{\neg, \wedge, \vee, \rightarrow\}$ and quantifiers $\{\forall, \exists\}$. Informally, a norm expresses that if, at some point, c_a holds, then the agent α is obliged to see to it that c_m is maintained at least until c_d holds; otherwise, α is obliged to see to it that c_r holds before the timeout c_t. For example, the norm

$$\langle Ag, driving(Ag), \neg cross\_red(Ag, LightID), \neg driving(Ag), fine\_paid(Value), time(500) \rangle$$

expresses that, when an agent Ag is driving, he is obliged to not cross the red traffic light $LightID$ until he is not driving; otherwise it has to pay $Value$ before the time 500 (words starting with upper case letters are variables).

The activation of the norms leads to the creation of norm instances, defined as follows.

Definition 4.2.2 (Norm instance) *Given a norm n and a substitution of variables* θ*, a norm instance is represented by* $n' = \langle \alpha', c'_a, c'_m, c'_d, c'_r, c'_t \rangle$ *s.t* α' *is an agent,* $c'_a = c_a\theta$*,* $c'_m = c_m\theta$*,* $c'_d = c_d\theta$*,* $c'_r = c_r\theta$*, and* $c'_t = c_t\theta$*.*

The set of all norm instances, noted N, is called normative state. It is defined as follows.

Definition 4.2.3 (Normative state) *The* normative state *is* $N = AS \cup VS \cup DS \cup FS$ *s.t. (i) AS is the set of active instances, (ii) VS is the set of violated instances, (iii) DS is the set of deactivated instances, and (iv) FS is the set of failed instances.*

As shown in Figure 12, a norm instance n' is activated as soon as its activation condition c'_a is satisfied, getting then into AS. If at some point the maintenance condition c'_m is not satisfied, the norm instance is violated, getting into VS. If the norm instance is active and the deactivation condition c'_d is satisfied, the norm instance gets deactivated (DS). If it is violated, either (i) fulfilling its reparation condition c'_r leads it to deactivated state (DS) or (ii) the occurrence of the timeout condition c'_t leads it to the failure state (FS).

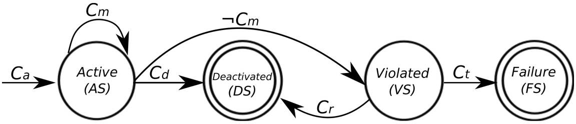


Figura 12 – Lifecycle of norm instances according to (PANAGIOTIDI; ÁLVAREZ-NAPAGAO; VÁZQUEZ-SALCEDA, 2013).

The predicates *active*, *viol*, *deactivated*, and *failed* are defined to check a norm instance with respect to the normative state N as follows:

$$N \models active(n') \text{ iff } n' \in AS \tag{4.1}$$

$$N \models viol(n') \text{ iff } n' \in VS \tag{4.2}$$

$$N \models deactivated(n') \text{ iff } n' \in DS \tag{4.3}$$

$$N \models failed(n') \text{ iff } n' \in FS \tag{4.4}$$

To express conditions over all the normative state, *n-formulae* $w_N \in W_n$ are defined following the grammar (4.5), having the semantics according to the expressions (4.1) to (4.4). Expressing such conditions over the normative state is useful, for example, to implement the norm chaining. A formula w_N can be used, for example, as the activation condition of a norm.

$$w_N ::= active(n')|viol(n')|deactivated(n')|failed(n')|w_N \wedge w_N|w_N \vee w_N|\bot|\top \tag{4.5}$$

In (PANAGIOTIDI; ÁLVAREZ-NAPAGAO; VÁZQUEZ-SALCEDA, 2013), a *normative monitor* is defined as a tuple $M_N = \langle \mathcal{N}, AS, VS, DS, FS, s \rangle$ where (i) $\mathcal{N}$ is the set of considered norms, (ii) AS, VS, DS, and FS are the sets of active, violated, deactivated, and failed norm instances, and (iii) s indexes the current state of the normative monitor. The transition system for a normative monitor M_N is $TS_{M_N} = \langle \Gamma_{M_N}, \rhd \rangle$ where Γ_{M_N} is the set of all possible configurations of the normative monitor and $\rhd \subseteq \Gamma_{M_N} \times \Gamma_{M_N}$ is a transition relation between configurations. The operational semantics of the normative monitor follows the transition rules (4.6) to (4.10).

$$\frac{\langle \alpha, c_a, c_m, c_d, c_r, c_t \rangle \in \mathcal{N} \quad c_a\theta \quad \neg c_d\theta}{M_N \rhd \langle \mathcal{N}, AS \cup \langle \alpha', c_a\theta, c_m\theta, c_d\theta, c_r\theta, c_t\theta \rangle, VS, DS, FS, s_{i+1} \rangle} \tag{4.6}$$

$$\frac{n' = \langle \alpha', c'_a, c'_m, c'_d, c'_r, c'_t \rangle \qquad n' \in AS \qquad \neg c'_m}{M_N \rhd \langle \mathcal{N}, AS \setminus n', VS \cup n', DS, FS, s_{i+1} \rangle} \tag{4.7}$$

$$\frac{n' = \langle \alpha', c'_a, c'_m, c'_d, c'_r, c'_t \rangle \qquad n' \in AS \qquad c'_d}{M_N \rhd \langle \mathcal{N}, AS \setminus n', VS, DS \cup n', FS, s_{i+1} \rangle} \tag{4.8}$$

$$\frac{n' = \langle \alpha', c'_a, c'_m, c'_d, c'_r, c'_t \rangle \qquad n' \in VS \qquad c'_r}{M_N \rhd \langle \mathcal{N}, AS, VS \setminus n', DS \cup n', FS, s_{i+1} \rangle} \tag{4.9}$$

$$\frac{n' = \langle \alpha', c'_a, c'_m, c'_d, c'_r, c'_t \rangle \qquad n' \in VS \qquad c'_t}{M_N \rhd \langle \mathcal{N}, AS, VS \setminus n', DS, FS \cup n', s_{i+1} \rangle} \tag{4.10}$$

The sets $\mathcal{N}$,AS, VS, DS, and FS are those of the M_N. The conditions c_a, c_d, c'_m, c'_d, c'_r, and c'_d are evaluated against the state of the world to manage the normative state N as illustrated in Figure 12. For example, by the transition rule (4.6), if the state of the world satisfies the activation condition but does not satisfies the deactivation condition of a norm – both of them under some substitution θ – then the monitor adds a norm instance $n' = \langle \alpha', c_a\theta, c_m\theta, c_d\theta, c_r\theta, c_t\theta \rangle$ into the set AS.

4.2.1.2 Aligning normative and constitutive representations

To link the representation of norms presented in Section 4.2.1.1 to the constitutive state presented in Chapter 3, we need to introduce status functions in the norms. Following the stated in Section 4.1, for a norm $n \in \mathcal{N}$, where $n = \langle \alpha, c_a, c_m, c_d, c_r, c_t \rangle$, we explicitly define that $\alpha \in \mathcal{A}_{\mathcal{F}}$, $c_a \in W_{\mathcal{F}} \cup W_N$, $c_m \in W_{\mathcal{F}} \cup W_N$, $c_d \in \mathcal{E}_{\mathcal{F}} \cup \mathcal{S}_{\mathcal{F}}$, $c_r \in \mathcal{E}_{\mathcal{F}} \cup \mathcal{S}_{\mathcal{F}}$, and $c_t \in W_{\mathcal{F}} \cup W_N$. Figure 13 shows the norms as conceived in (PANAGIOTIDI; ÁLVAREZ-NAPAGAO; VÁZQUEZ-SALCEDA, 2013) using the status functions of Figure 10 to specify that (i) the *mayor* is obliged to evacuate secure zones and (ii) *firefighters* are obliged to evacuate insecure zones.

$1: \langle \texttt{mayor}, \texttt{secure(downtown)}, \texttt{secure(downtown)}, \texttt{evacuate(downtown)}, \bot, \neg\texttt{secure(downtown)} \rangle$
$2: \langle \texttt{firefighter}, \texttt{insecure(downtown)}, \texttt{insecure(downtown)}, \texttt{evacuate(downtown)}, \bot,$
$\neg\texttt{insecure(downtown)} \rangle$

Figura 13 – Norms using status functions

While norms refer to agent-status functions (i.e. $\alpha \in \mathcal{A}_{\mathcal{F}}$), their instances prescribe the behaviour of the concrete agents acting in the environment. The obligation of an agent $a_X \in A_X$ to follow a norm instance n' is conditioned by its carry of the status function α as prescribed in the norm n. As detailed later in the expressions (4.12) to (4.15), to check this condition considering individually the agents, norm instances must record both the agent to whom the instance is directed and the status function carried by that agent when the instance was created. Thus, in an instance $n' = \langle \alpha', c'_a, c'_m, c'_d, c'_r, c'_t \rangle$, we consider $\alpha' = (a_X, \alpha)$ where $a_X \in A_X$ points to the concrete agent targeted by the norm instance and $\alpha \in \mathcal{A}_{\mathcal{F}}$ is the status function carried by that agent when the instance was created.

4.2.1.3 Coupling normative and constitutive dynamics

Having defined how normative and constitutive representations are linked, this section explains how the dynamics of the normative and constitutive states are coupled.

Activation. Given a normative specification $\mathcal{N}$, a constitutive state F and a normative state N, the set of norms instances to be created is given by the function *activated* defined below:

$$\begin{aligned} activated(\mathcal{N}, F, N) = \{n' | \exists\theta\exists\langle \alpha, c_a, c_m, c_d, c_r, c_t \rangle \in \mathcal{N} : \\ F \cup N \models c_a\theta \wedge (a_X \textbf{ is } \alpha\theta) \wedge n' \notin AS\} \end{aligned} \tag{4.11}$$

$$\text{s.t.} \quad n' = \langle (a_X, \alpha\theta), c_a\theta, c_m\theta, c_d\theta, c_r\theta, c_t\theta \rangle$$

The creation of norm instances is conditioned by the constitutive and normative states satisfying the activation condition c_a for some substitution θ (i.e. $F \cup N \models c_a\theta$). The evaluation of c_a with respect to N follows the expressions (4.1) to (4.4). Its evaluation with respect to F follows the expressions (3.2) and (3.3). By the function *activated*, a norm directed to an agent-status function α produces an instance for *every* concrete agent a_X carrying α. For example, considering the specification in Figure 10, if the agents *bob* and *tom* carry the status function of *firefighter* (i.e. $\{\langle bob, firefighter \rangle, \langle tom, firefighter \rangle\} \subseteq A_F$) and the *downtown* is in emergency phase of crisis, being thus insecure (i.e. $\langle security\_phase(downtown, emergency), insecure(downtown) \rangle \in S_F$), then (i) $F \models insecure(downtown)$, (ii) $F \models bob \textbf{ is } firefighter$, and (iii) $F \models$

tom **is** *firefighter*. Thus, the following instances of the norm 2 are created:

$$\langle (bob, firefighter), insecure(downtown), insecure(downtown),\\ evacuate(downtown), \bot, \neg insecure(downtown)\rangle$$

$$\langle (tom, firefighter), insecure(downtown), insecure(downtown),\\ evacuate(downtown), \bot, \neg insecure(downtown)\rangle$$

Deactivation. Deactivations are considered separately according to the nature of the deactivation condition (event or state). The functions $f\text{-}deactivated^e$ and $f\text{-}deactivated^s$ deal respectively with deactivations of active instances conditioned by events and by states.

$$f\text{-}deactivated^e(F,N) = \{n' | \exists (e_X, a_X) \in E_X : n' \in AS \wedge c'_d \in \mathcal{E_F} \wedge \\ F \models ((e_X, a_X) \textbf{ is } c'_d \vee \neg(a_X \textbf{is } \alpha)) \wedge F \cup N \models c'_m\} \tag{4.12}$$

$$f\text{-}deactivated^s(F,N) = \{n' | n' \in AS \wedge c'_d \in \mathcal{S_F} \wedge \\ F \models (c'_d \vee \neg(a_X \textbf{is } \alpha)) \wedge F \cup N \models c'_m\} \tag{4.13}$$

$$\text{s.t.} \quad n' = \langle (a_X, \alpha), c'_a, c'_m, c'_d, c'_r, c'_t\rangle$$

The function $f\text{-}deactivated^e$ captures the notion of events as being considered at the individual agent level. The obligation of an agent a_X with respect to the occurrence in the environment of an event that counts as the event-status function c'_d is only fulfilled when c'_d is assigned to an event e_X really produced by the agent a_X. This is expressed by the element $F \models ((e_X, a_X) \textbf{ is } c'_d)$, evaluated according to the Expression (3.3). By the function $f\text{-}deactivated^s$ an agent fulfils an obligation to achieve a state when it *sees to it* that such state holds, no matter by whom it has been produced. This achievement is detected when there is an assignment to the state-status function c'_d, evaluated according to the Expression (3.2).

The functions $f\text{-}deactivated^e$ and $f\text{-}deactivated^s$ capture the idea of norm instances being directed to the concrete agents but being conditioned by the agent-status function assignments. If an instance is assigned to the agent a_X because it carries the agent-status function α, then it is deactivated if a_X ceases to carry α. For example, we can imagine that the agent *bob* is obliged to evacuate the downtown because it carries the agent-status function *firefighter*. As the obligation was specified to the *firefighter* rather than to *bob*, it should be deactivated as soon *bob* looses this function.

While active norm instances are deactivated when the deactivation condition c'_d is satisfied, violated instances are deactivated by the satisfaction of the repair condition c'_r. Deactivations by reparation of violated instances are also considered at the individual agent level when they are conditioned by events (function *r-deactivated*^{e}). Reparations conditioned by states are achieved when the agents see to them that such state holds (function *r-deactivated*^{s}). Different of deactivations of active instances, the maintenance condition is not considered in the reparations of violated ones. An instance, to be repaired, must be in the violated state, reached when the maintenance condition c'_m ceased to hold in the past. If the c'_m holds while the reparation condition of a violated instance is reached, it has started to hold again while the instance was violated, having thus no influence on such instance.

$$\begin{aligned} \textit{r-deactivated}^e(F,N) = \{n' | \exists (e_X, a_X) \in E_X : n' \in VS \wedge c'_r \in \mathcal{E}_{\mathcal{F}} \wedge \\ F \models ((e_X, a_X)\ \mathbf{is}\ c'_r \vee \neg (a_X \mathbf{is}\ \alpha))\} \end{aligned} \tag{4.14}$$

$$\begin{aligned} \textit{r-deactivated}^s(F,N) = \{n' | n' \in VS \wedge c'_r \in \mathcal{S}_{\mathcal{F}} \wedge \\ F \models (c'_r \vee \neg (a_X \mathbf{is}\ \alpha))\} \end{aligned} \tag{4.15}$$

$$\text{s.t.} \quad n' = \langle (a_X, \alpha), c'_a, c'_m, c'_d, c'_r, c'_t \rangle$$

Violation. Active norm instances are considered violated when the constitutive and normative states do not satisfy the maintenance condition (function *violated* below).

$$\textit{violated}(F,N) = \{n' | n' \in AS \wedge F \cup N \not\models c'_m\} \tag{4.16}$$

$$\text{s.t.} \quad n' = \langle (a_X, \alpha), c'_a, c'_m, c'_d, c'_r, c'_t \rangle$$

Failure. An instance is failed if (i) it is violated and (ii) the current constitutive and normative states satisfy the timeout condition (function *failed* below).

$$\textit{failed}(F,N) = \{n' | \exists \theta : n' \in VS \wedge F \cup N \models \ c_t{}'\} \tag{4.17}$$

$$\text{s.t.} \quad n' = \langle (a_X, \alpha), c'_a, c'_m, c'_d, c'_r, c'_t \rangle$$

Monitoring norms based on the constitutive state. The original operational semantics of the norm monitor proposed in (PANAGIOTIDI; ÁLVAREZ-NAPAGAO; VÁZQUEZ-SALCEDA, 2013) considers that the lifecycle of the norm instances evolves based on the satisfaction of the conditions c_a, c_d, c_m, c_r, and c_t. To base the regulation on the constitutive state, it is necessary to consider also other conditions captured by the functions 4.11 to 4.17. For this reason, to base the operational semantics of the norm monitor proposed in (PANAGIOTIDI; ÁLVAREZ-NAPAGAO; VÁZQUEZ-SALCEDA, 2013) on the SAI constitutive state, we redefine below the transition rules presented in Section 4.2.1.1.

$$\frac{n' \in activated(\mathcal{N},F,N) \qquad n' \notin f\text{-}deactivated^{e}(F,N) \cup f\text{-}deactivated^{s}(F,N)}{M_N \triangleright \langle \mathcal{N}, AS \cup n', VS, DS, FS, s_{i+1} \rangle} \tag{4.18}$$

$$\frac{n' \in AS \qquad n' \in violated(F,N)}{M_N \triangleright \langle \mathcal{N}, AS \setminus n', VS \cup n', DS, FS, s_{i+1} \rangle} \tag{4.19}$$

$$\frac{n' \in AS \qquad n' \in f\text{-}deactivated^{e}(F,N) \cup f\text{-}deactivated^{s}(F,N)}{M_N \triangleright \langle \mathcal{N}, AS \setminus n', VS, DS \cup n', FS, s_{i+1} \rangle} \tag{4.20}$$

$$\frac{n' \in VS \qquad n' \in r\text{-}deactivated^{e}(F,N) \cup r\text{-}deactivated^{s}(F,N)}{M_N \triangleright \langle \mathcal{N}, AS, VS \setminus n', DS \cup n', FS, s_{i+1} \rangle} \tag{4.21}$$

$$\frac{n' \in VS \qquad n' \in failed(F,N)}{M_N \triangleright \langle \mathcal{N}, AS, VS \setminus n', DS, FS \cup n', s_{i+1} \rangle} \tag{4.22}$$

4.2.2 Coupling NPL norms

This section describes how the norms following the NPL normative model proposed in (HÜBNER; BOISSIER; BORDINI, 2010, 2011) can base their regulation on the SAI constitutive state. Section 4.2.2.1 describes the normative model, Section 4.2.2.2 describes how the particular representations of the normative model are aligned with the constitutive representations of SAI, and Section 4.2.2.3 defines how the particular dynamics of the normative model is animated based on the constitutive dynamics.

4.2.2.1 The normative model

In NPL, a norm has the form `norm` *id* : $\varphi \rightarrow \psi$ where *id* is a unique identifier of the norm; φ is a formula defining the activation condition of the norm; and ψ is the consequence of the activation of the norm. There are two types of consequences:

1. `fail`(r), used for regimented norms. In this case, the activation condition represents an undesirable state that should be regimented.[2] It is up to the normative platform to handle the failure to achieve again a consistent state. The element r represents the reason for the failure.
2. `obligation`(a,r,g,d), representing an obligation for the agent a. Argument r is the reason for the obligation; g is the goal to be achieved; and d is the deadline to fulfil the obligation.

Figure 14 shows NPL norms as using the status functions of Figure 10 to specify that (i) the *mayor* is obliged to evacuate secure zones ($n1$) and (ii) *firefighters* are obliged to evacuate insecure zones ($n2$).

```
norm n1: secure(Zone)
 -> obligation(mayor,n1,evacuate(Zone),‘now’+‘4 hours’).

norm n2: insecure(Zone)
 -> obligation(firefighter,n2,evacuate(Zone),‘now’+‘4 hours’).
```

Figura 14 – Examples of NPL norms

An obligation is created, getting *active* when the activation condition φ is satisfied. An active obligation can become (i) *fulfilled*, when the goal g is achieved before the deadline d; (ii) *unfulfilled*, when the deadline d is satisfied before the fulfilment of g; or (iii) *inactive*, when the activation condition φ ceases to hold (cf. Figure 15).

The state of a NPL normative system is a tuple $\langle F,N,s,OS,t\rangle$ where (i) F is the set of facts considered in the evaluation of the norms, (ii) N is a set of norms, (iii) s is the state of normative system, that can be sound, denoted by $\top$ or failure, denoted by $\bot$, (iv) OS is the state of the obligations s.t. each obligation $os \in OS$ is a pair $\langle o,ost\rangle$ where o is an obligation in the state $ost \in \{\mathbf{active},\mathbf{fulfilled},\mathbf{unfulfilled},\mathbf{inactive}\}$, and (v) t is the current time considered by the normative management platform.

---

[2] Regimentation is the strategy to lead individuals to behave as expected by preventing them to violate the norms (GROSSI; GABBAY; TORRE, 2010).

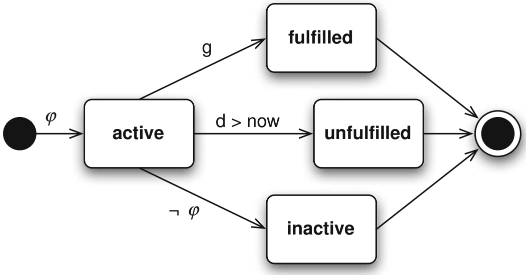


Figura 15 – Lifecycle of obligations in NPL (HÜBNER; BOISSIER; BORDINI, 2011)

4.2.2.2 Aligning normative and constitutive representations

For a norm $n \in \mathcal{N}$, where $n = id : \varphi \rightarrow \psi$, we first explicitly define that $\varphi \in W_{\mathcal{F}}$ as the activation condition of a norm is evaluated with respect to the whole constitutive state. Failures raised when $\psi = \texttt{fail}(r)$ are internally handled by the NPL engine and are not evaluated against the constitutive state. When the consequence is an $\texttt{obligation}(a, r, g, d)$, we explicitly define that a in $\mathcal{A}_{\mathcal{F}}$, $g \in \mathcal{E}_{\mathcal{F}} \cup \mathcal{S}_{\mathcal{F}}$, and $r \in \mathcal{W}_{\mathcal{F}}$. The deadline d is expressed as time, that is managed by the normative platform, being thus not related to the status functions.

For a norm $n = id : \varphi \rightarrow \psi$ where $\psi = \texttt{obligation}(a, r, g, d)$, the satisfaction of the activation condition φ under a substitution θ creates instances of $\texttt{obligation}(a, r, g, d)$. An instance of $\texttt{obligation}(a, r, g, d)$ is represented as $\langle a', r', g', o' \rangle$ where $r' = r\theta$, $g' = g\theta$, and $o_d = d\theta$. While norms in NPL specify obligations considering the agent-status functions, the activation of the norms creates obligations to the agents carrying the agent-status functions. An $\texttt{obligation}(a, r, g, d)$ must be followed by an agent a_X if it carries the status function a as prescribed in the norm. Thus, in an instance $o = \langle a', r', g', o_d \rangle$ of an $\texttt{obligation}(a, r, g, d)$, we consider $a' = (a_X, a)$ where a_X points to the concrete agent targeted by the norm instance and $a \in \mathcal{A}_{\mathcal{F}}$ is the status function carried by that agent when the instance was created.

4.2.2.3 Coupling normative and constitutive dynamics

In (HÜBNER; BOISSIER; BORDINI, 2011), norms are evaluated with respect to a set of facts F. When the NPL dynamics is based on the SAI constitutive state, this set of facts F is the set of status function assignments composing the SAI constitutive state. As said before, norm activations can raise failures and obligations. Failures require regimentation, that concerns to the internal management of NPL, that is to say, it is not related to the constitutive state. Our focus here is on the obligations, whose lifecycle illustrated in Figure 15 evolves based on the constitutive state.

Based on the original NPL operational semantics (HÜBNER; BOISSIER; BORDINI, 2011), we describe in the following how obligations, as conceived by NPL, are activated, fulfilled, deactivated, and violated when norms are based on the SAI constitutive state. In the transition rules 4.23 to 4.26 we always consider that the normative state evolves based on the current constitutive state F. A reference to F implicitly refers to some step i of the history of the constitutive state (i.e. F^i).

Activation. The instantiation of obligations is conditioned by the constitutive state satisfying the activation condition φ for some substitution θ (i.e. $F \models \varphi\theta$). The evaluation of $\varphi\theta$ with respect to F follows the expressions 3.2 and 3.3. By the transition rule 4.23, an obligation directed to an agent-status function a produces an instance for *every* concrete agent a_X carrying a. An agent a_X is targeted by an obligation because he carries the status function a and, thus, a_X **is** a is part of the reason r for the obligation to hold. The NPL semantics does not allow the creation of another instance of an active obligation with a different deadline (HÜBNER; BOISSIER; BORDINI, 2011). The notation $\overset{obl}{=}$ is used for equality of obligations ignoring the deadline in the comparison.

$$\frac{\begin{array}{c}\texttt{norm}\, id : \varphi \rightarrow \psi \in N \qquad \psi = o \qquad F \models \varphi\theta \wedge a_X\ \mathbf{is}\ a \qquad d\theta > t \\ \neg\exists\langle o', ost\rangle \in OS : (o' \overset{obl}{=} o\theta \wedge ost \neq \mathbf{active})\end{array}}{\langle F, N, \top, OS, t\rangle \rightarrow \langle F, N, \top, OS \cup \langle o\theta, \mathbf{active}\rangle, t\rangle} \tag{4.23}$$

s.t. $o = \texttt{obligation}\langle a, r, g, d\rangle$ and $o\theta = \langle (a_X, a), r\theta \wedge a_x\ \mathbf{is}\ a, g\theta, d\theta)\rangle$

For example, considering the specification in Figure 10, if the agents *bob* and *tom* carry the status function *firefighter* (i.e. $\{\langle bob, firefighter\rangle, \langle tom, firefighter\rangle\} \subseteq A_F$) and the *downtown* is in emergency phase of crisis, being thus insecure (i.e.

$\langle security\_phase(downtown, emergency), insecure(downtown)\rangle \in S_F$), then (i) $F \models insecure(downtown)$, (ii) $F \models bob\,\textbf{is}\,firefighter$, and (iii) $F \models tom\,\textbf{is}\,firefighter$. Thus, considering the norm n2 shown in Figure 14, the following obligations are created:

$$\langle (bob, firefighter), insecure(downtown) \wedge bob\,\textbf{is}\,firefighter, evacuate(downtown), o_d\rangle$$
$$\langle (tom, firefighter), insecure(downtown) \wedge tom\,\textbf{is}\,firefighter, evacuate(downtown), o_d\rangle$$

Deactivation. The state of an active obligation o should be changed to **inactive** if the reason for the obligation ceases to hold in the current system state reflected in F(HÜBNER; BOISSIER; BORDINI, 2011). This is expressed by $F \not\models r'$ in the transition rule 4.24.

$$\frac{os \in OS \quad os = \langle o, \textbf{active}\rangle \quad F \not\models r'}{\langle F, N, \top, OS, t\rangle \rightarrow \langle F, N, \top, (OS \setminus \{os\}) \cup \{\langle o, \textbf{inactive}\rangle\}, t\rangle} \tag{4.24}$$

$$\text{s.t. } o = \langle (a_X, a), r', g', d')\rangle$$

The transition rule 4.24 implicitly captures the idea of obligations being directed to the concrete agents but being conditioned by the agent-status function assignments. Remember that, if $\langle (a_X, a), r', g', d'\rangle$ is an obligation produced by `norm` $id : \varphi \rightarrow \langle a, r, g, d\rangle$, then $r' = r\theta \wedge a_X \textbf{is}\, a$ (cf. transition rule 4.23). If an instance is assigned to the agent a_X because it carries the agent-status function a, then it is deactivated if a_X ceases to carry a (i.e. if $F \not\models a_X \textbf{is}\, a$). For example, we can imagine that the agent *bob* is obliged to evacuate the downtown because he carries the agent-status function of firefighter. As the obligation was directed to the *firefighter* rather than to *bob*, it should be deactivated as soon *bob* looses this function.

Fulfilment. Fulfilments are considered separately according to the nature of the goal to be achieved (event or state). The transition rules 4.25 and 4.26 deal respectively with fulfilments of active instances conditioned by events and by states.

$$\frac{\begin{array}{c} os \in OS \quad os = \langle o, \textbf{active}\rangle \quad g' \in \mathcal{E}_{\mathcal{F}} \\ \exists (e_X, a_X) \in E_X : F \models (e_X, a_X) \textbf{is}\, g' \end{array}}{\langle F, N, \top, OS, t\rangle \rightarrow \langle F, N, \top, (OS \setminus \{os\}) \cup \{\langle o, \textbf{fulfilled}\rangle\}, t\rangle} \tag{4.25}$$

$$\frac{os \in OS \quad os = \langle o, \textbf{active} \rangle \quad g' \in \mathcal{S}_{\mathcal{F}} \quad F \models g'}{\langle F, N, \top, OS, t \rangle \rightarrow \langle F, N, \top, (OS \setminus \{os\}) \cup \{\langle o, \textbf{fulfilled} \rangle\}, t \rangle} \tag{4.26}$$

$$\text{s.t. } o = \langle (a_X, a'), r', g', o_d) \rangle$$

The transition rule 4.25 captures the notion of events as being considered at the individual agent level. The obligation of an agent a_X with respect to the occurrence in the environment of an event that counts as the event-status function g' is only fulfilled when g' is assigned to an event e_X really produced by the agent a_X. This is expressed by the element $F \models ((e_X, a_X) \textbf{ is } g')$, evaluated according to the Expression (3.3). By the transition rule 4.26, an agent fulfils an obligation to achieve a state when it *sees to it* that such state holds, no matter by whom it has been produced. This achievement is detected when there is an assignment to the state-status function g', evaluated according to the Expression (3.2).

Unfulfillment. A NPL obligation moves from active to unfulfilled if the deadline is already past. Deadlines in NPL are checked considering a discrete, linear, notion the time, internally managed by the normative engine. Thus, conditions to unfulfilments are not constituted from the environmental state.

Failures and Regimentation. In NPL, when the facts F reflect an undesirable state, the normative state goes to a failure one. The transition rule 4.27, defining how the normative state is moved from a consistent ($\top$) to a failure ($\perp$) state, is the same as the original one introduced in (HÜBNER; BOISSIER; BORDINI, 2011). As the activation condition of the norms is evaluated with respect to the whole constitutive state ($F \models \varphi$), failure normative states are particular undesirable *constitutive* states.

$$\frac{n \in N \quad F \models \varphi \quad n_\psi = \texttt{fail(\_)}}{\langle F, N, \top, OS, t \rangle \rightarrow \langle F, N, \perp, OS, t \rangle} \tag{4.27}$$

When failure normative states are produced, the NPL engine rolls back the facts in F to the previous consistent state. When NPL is coupled to SAI, the set of facts that should be rolled back is the constitutive state. But, in this case, the rolled back constitutive state could become inconsistent with respect to the environment. For example, if $\langle x, y, \top, true \rangle \in \mathcal{C}$, $y \in \mathcal{S}_{\mathcal{F}}$, and $x \in S_X$, then $\langle x, y \rangle \in S_F$. If y produces a failure normative state and the constitutive state is rolled back, then $\langle x, y \rangle$ is removed from S_F. This leads to an inconsistency as, considering the environmental state and the given constitutive rule, x should count as y.

This issue seems to be related to the very notion of regimentation and to its implementation in NPL. Regimentation is as *preventive* strategy: the agents are prevented to produce undesirable states (TORRE et al., 2004; FIROZABADI; TORRE, 1998). Regimentation is thus a way to impose the agents' expected behaviours through hard constraints in a restrictive environment (TORRE et al., 2004). NPL strategy, however, is not preventive: agents acting in the environment can produce undesirable states that are rolled back in the normative representation of the facts. That is to say, while regimentation usually consists of preventing the agents to lead the world to undesirable states, NPL strategy allows the agents to do that and the normative engine is supposed to be able to undo the undesirable states.

The NPL strategy to deal with undesirable states seems to mix the notions of regulation and of institutional reality (even the later is not explicit in the NPL model). A norm whose the consequence is a failure can be seen as stating that certain conditions (represented by the activation condition) count as (i.e. are seen from the institutional perspective) as an undesirable state that the normative platform must deal with. Analysing and disentangling such a mixing of regulation and constitution in normative models is a future work. In this direction, a possible strategy to regulate undesirable states in NPL would be to explicitly take `fail(`r`)` as a state-status function to be constituted by undesirable states. This status function could be part of the activation condition of the norms whose consequence is an obligation. Thus, failure states would produce obligations to the agents to act in the environment producing environmental facts that bring the normative state consistent again.

4.3 EXAMPLES

To illustrate the evolution of the normative regulation based on the constitutive state, we consider the scenario introduced in Section 3.4, where five steps of the environmental dynamics produce changes in the constitutive state, as summarized in Table 2. To illustrate norms following different normative models being based on the same constitutive state, we consider both the norms shown in Figure 13, that follow the model of (PANAGIOTIDI; ÁLVAREZ-NAPAGAO; VÁZQUEZ-SALCEDA, 2013) and the norms shown in Figure 14, that follow the NPL model. Norms in both the sets prescribe the same behaviour: the *mayor* is obliged to evacuate the *downtown* when it is *secure* while the *firefighter* is obliged to evacuate it when it is *inse-*

cure.

The evolution of the normative states produced by both the sets of norms, described below, is summarized in tables 3 and 4:

- **Step 1.** As the downtown is considered secure and *bob* considered as *mayor*, the agent is obliged to evacuate that zone;
- **Step 2**. From the institutional perspective, *bob* has evacuated the downtown, fulfilling the previously created obligation;
- **Step 3**. The downtown becomes insecure from the institutional perspective. Thus, new obligations are created directed to the *firefighters*;
- **Step 4**. *Tom* puts the *launch_object* perform actions that count as the evacuation of the downtown, fulfilling their obligations.
- **Step 5**. From the institutional perspective, the downtown is again secure. As *ana* did not evacuate the downtown while it was insecure, its obligation becomes inactive.

Tabela 3 – Evolution of the normative state produced by norms according to (PANAGIOTIDI; ÁLVAREZ-NAPAGAO; VÁZQUEZ-SALCEDA, 2013)

Step	Normative state
1	$AS = \{\langle(bob,mayor),secure(downtown),secure(downtown),evacuate(downtown),\bot,\neg secure(downtown)\}\rangle$
2	$DS = \{\langle(bob,mayor),secure(downtown),secure(downtown),evacuate(downtown),\bot,\neg secure(downtown)\rangle\}$
3	$AS = \{\langle(tom,firefighter),insecure(downtown),insecure(downtown),evacuate(downtown),\bot,\neg insecure(downtown)\rangle,$ $\langle(jim,firefighter),insecure(downtown),insecure(downtown),evacuate(downtown),\bot,\neg insecure(downtown)\rangle,$ $\langle(ana,firefighter),insecure(downtown),insecure(downtown),evacuate(downtown),\bot,\neg insecure(downtown)\rangle\}$ $DS = \{\langle(bob,mayor),secure(downtown),secure(downtown),evacuate(downtown),\bot,\neg secure(downtown)\rangle\}$
4	$AS = \{\langle(ana,firefighter),insecure(downtown),insecure(downtown),evacuate(downtown),\bot,\neg insecure(downtown)\rangle\}$ $DS = \{\langle(bob,mayor),secure(downtown),secure(downtown),evacuate(downtown),\bot,\neg secure(downtown)\rangle,$ $\langle(tom,firefighter),insecure(downtown),insecure(downtown),evacuate(downtown),\bot,\neg insecure(downtown)\rangle,$ $\langle(jim,firefighter),insecure(downtown),insecure(downtown),evacuate(downtown),\bot,\neg insecure(downtown)\rangle\}$
5	$DS = \{\langle(bob,mayor),secure(downtown),secure(downtown),evacuate(downtown),\bot,\neg secure(downtown)\rangle,$ $\langle(tom,firefighter),insecure(downtown),insecure(downtown),evacuate(downtown),\bot,\neg insecure(downtown)\rangle,$ $\langle(jim,firefighter),insecure(downtown),insecure(downtown),evacuate(downtown),\bot,\neg insecure(downtown)\rangle\}$ $VS = \{\langle(ana,firefighter),insecure(downtown),insecure(downtown),evacuate(downtown),\bot,\neg insecure(downtown)\rangle\}$

The norms are designed, in both the considered models, to prescribe the same behaviour. As we can observe, the resulting regulation is similar. But the point here is not to show the norms producing the same regulation. Rather, the point is to show that norms following different models can be part of SAI and, having their particular regulation evolving based on the same constitutive state. The different sets of norms could specify different, even conflicting behaviours. But, independent of the behaviour being prescribed, the evolution of all the norms is based on the same components of the institutional reality, that are grounded in the environment in the same way.

Tabela 4 – Evolution of the normative state produced by NPL norms

Step	Normative state
1	$OS = \{\langle \texttt{obligation(}bob, n1, evacuate(downtown), \text{`}now' + \text{`}4\ hours'\texttt{)}, \textbf{active}\rangle\}$
2	$OS = \{\langle \texttt{obligation(}bob, n1, evacuate(downtown), \text{`}now' + \text{`}4\ hours'\texttt{))}, \textbf{fulfilled}\rangle\}$
3	$OS = \{\langle \texttt{obligation(}bob, n1, evacuate(downtown), \text{`}now' + \text{`}4\ hours'\texttt{))}, \textbf{fulfilled}\rangle,$ $\langle \texttt{obligation(}tom, n2, evacuate(downtown), \text{`}now' + \text{`}4\ hours'\texttt{)}, \textbf{active}\rangle,$ $\langle \texttt{obligation(}jim, n2, evacuate(downtown), \text{`}now' + \text{`}4\ hours'\texttt{)}, \textbf{active}\rangle,$ $\langle \texttt{obligation(}ana, n2, evacuate(downtown), \text{`}now' + \text{`}4\ hours'\texttt{)}, \textbf{active}\rangle\}$
4	$OS = \{\langle \texttt{obligation(}bob, n1, evacuate(downtown), \text{`}now' + \text{`}4\ hours'\texttt{))}, \textbf{fulfilled}\rangle,$ $\langle \texttt{obligation(}tom, n2, evacuate(downtown), \text{`}now' + \text{`}4\ hours'\texttt{)}, \textbf{fulfilled}\rangle,$ $\langle \texttt{obligation(}jim, n2, evacuate(downtown), \text{`}now' + \text{`}4\ hours'\texttt{)}, \textbf{fulfilled}\rangle,$ $\langle \texttt{obligation(}ana, n2, evacuate(downtown), \text{`}now' + \text{`}4\ hours'\texttt{)}, \textbf{active}\rangle\}$
5	$OS = \{\langle \texttt{obligation(}bob, n1, evacuate(downtown), \text{`}now' + \text{`}4\ hours'\texttt{))}, \textbf{fulfilled}\rangle,$ $\langle \texttt{obligation(}tom, n2, evacuate(downtown), \text{`}now' + \text{`}4\ hours'\texttt{)}, \textbf{fulfilled}\rangle,$ $\langle \texttt{obligation(}jim, n2, evacuate(downtown), \text{`}now' + \text{`}4\ hours'\texttt{)}, \textbf{fulfilled}\rangle,$ $\langle \texttt{obligation(}ana, n2, evacuate(downtown), \text{`}now' + \text{`}4\ hours'\texttt{)}, \textbf{inactive}\rangle\}$

4.4 CONCLUSIONS

It worths to recall, at this point, the main question to be answered by this thesis: *how to represent the institutional reality arising from the environment to base the whole regulation on in artificial institutions?* The design of the institutional reality was addressed in Chapter 3. This chapter, on its turn, addressed the coupling of the regulation, independent of how it is conceived, with the institutional reality. Questions to be answered are related to whether and how norms proposed in the literature can base their regulation on the constitutive state.

Section 4.1 shows that norms in general, independent of the model they follow, can be *expressed* in terms of status functions. The institutional reality in SAI – i.e. the constitutive state – provides a common vocabulary, composed of the status functions, in terms of which all the norms of an institution can be expressed. But more than provide the set of words to be used to write norms, the institutional reality as conceived by SAI provides semantics to the institutional vocabulary through the typing of the status functions. It is possible, thus, to establish clear relations between the components of the norms and the components of the institutional reality (cf. Section 4.1). For example, it is possible to define that both the "α" of Panagiotidi et *al.* and the "a" of the NPL norms refer to the same kind of element, namely agent-status functions. Norms, thus, are consistent with institutional reality and also with the environment under regulation. For instance, if *firefighter* is an agent-status function and *evacuate* is an event-status function, *evacuate* cannot be the attribute and *firefighter* cannot be the aim of a norm. Such consistency does not depends neither on the normative model nor on the normative specification. Rather, it is provided by the proper link between the components of the norms and the different kinds of status functions.

Basing the normative regulation on the constitutive state is not just about to specify norms through status functions. It is necessary to define how to manage norms based on an *interpretation* of the environment to regulate the elements *under such interpretation*. As described in Section 4.2, we addressed this point in two steps: (i) aligning constitutive and normative representations and (ii) defining how lifecycle of the norm instances evolves according to the constitutive state. These two steps can be seen as a replicable strategy (kind of "method") to couple different normative models with the constitutive state. It was applied, in Section 4.2 to two normative models, considering their particular semantics.

Coupling norms with the constitutive state involves some decisions. First, it is necessary to define how the elements abstracted under the constitution are considered in the management of the normative state. Our proposed couplings explicitly define that (i) regarding to the addressee (or the ***A****ttribute*), norms govern all the agents under the same constitution of agent-status function while (ii) the *a****I****m* and the ***C****onditions*, differently, point to (at least) a single constitution of event- and state-status function. For example, considering a norm stating that firefighters are obliged to evacuate an insecure zone (figures 13 and 14 – norm 2), we can imagine a situation where many agents count as firefighter and two events count as an evacuation (Figure 10 – constitutive rules 3 and 4). When instantiated, this norm stands to all the agents counting as *firefighter* but its fulfilment requires that every firefighter produces at least one event interpreted as *evacuation*. They can either put a tangible on the table or send a message (Figure 10 – constitutive rules 3 and 4).

A second decision we took is to consider a norm instance as deactivated when the responsible agents are no longer carrying the target status function (expressions 4.12 to 4.15 and 4.24). This is because, as the normative regulation is based on the constitutive state, the expected agents' behaviour is attached to the status functions instead of to the agents themselves. But other coupling approaches can be conceived where, for example, obligations and prohibitions remain active even if the agent-status functions are revoked. These decisions are related to the management of the social meanings of the agents in a society that, as noted in (TESSOP, 2011), is a complex question that can be addressed in different ways.

We have, thus, a model to represent the institutional reality in MAS (cf. Chapter 3) and, as described in this chapter, the directions to couple the regulation provided by norms, independent of a specific normative model, in this reality. The next chapter describes an architecture to deploy institutions following this conception, having elements to represent and manage the institutional reality as well as to incorporate different normative engines, whose norms possibly follow different models. Application examples to illustrate SAI regulating MAS are described in chapters 6 and 7.

5 ARCHITECTURE OF AN INSTITUTIONAL PLATFORM

Chapters 3 and 4 provide a conceptual apparatus to represent and animate artificial institutions where the institutional reality is constituted from the environment to base the normative regulation. From the presented concepts, this chapter aims at advancing on practical issues in institutional design and programming. Regarding the objectives of this thesis, stated in Section 1.4, this chapter contributes (i) to define the architecture of an interpreter for the proposed language and (ii) to define the required machinery to base normative regulation following different normative models in the proposed interpreter.

To achieve these objectives, this chapter proposes an architecture for an institutional platform to enable the deployment of institutions following the SAI model in MAS. The proposed architecture includes (i) an interpreter, that, by interpreting constitutive specifications, manages the constitutive state of the institution and (ii) the required elements to insert different normative engines in the institution. In the following, Section 5.1 introduces the proposed architecture; Section 5.2 describes the elements that manage the constitutive state; Section 5.3 describe the elements that enable to add different normative engines to the same institutional platform; and Section 5.4 describes an example where the proposed architecture is integrated with existing environmental and normative platforms.

5.1 INTRODUCTION

This chapter proposes the architecture of an institutional platform, defining their components, the relations among them, and their interaction with the elements external to the institution. Thus, different implementations following the proposed architecture are possible. It is important also to remark that the architecture proposed in this chapter is one among the many different architectures that could be proposed to deploy SAI institutions.

The general view of the proposed architecture is shown in the component diagram of Figure 16, that follows the UML notation (BOOCH; RUMBAUGH; JACOBSON, 2005). The institutional platform is, itself, conceived as a component to be inserted in a broader system. It is represented in Figure 16 by the component *SAI_Platform*. In SAI, an institution is conceived to animate both the institutional reality and the normative regulation. This feature is captured by the *SAI_Platform*, that is composed of two kinds of components: the *Constitutive_Engine* and some *Normative_Engine*, presented in

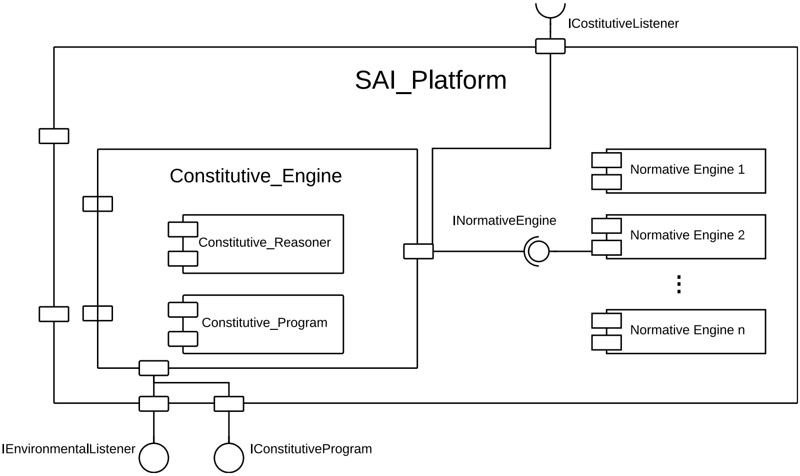


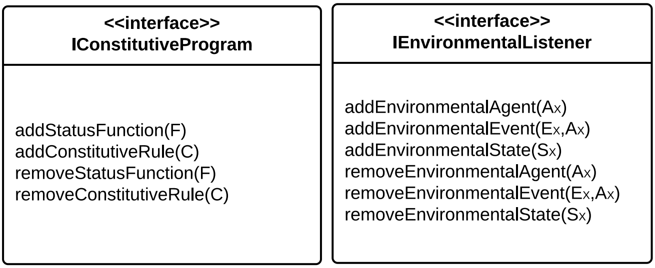
Figura 16 – Component diagram of SAI platform

sections 5.2 and 5.3 respectively.

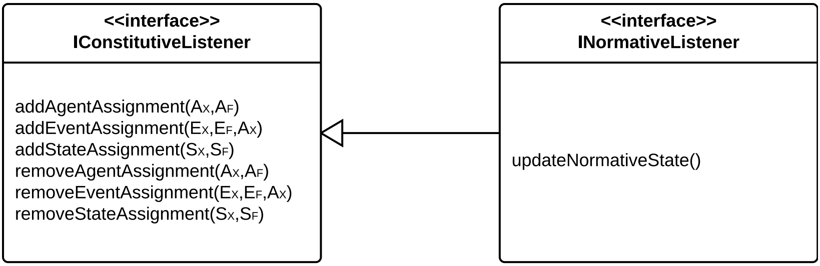
5.2 MANAGING THE CONSTITUTIVE STATE: THE *CONSTITUTIVE_ENGINE*

The *Constitutive_Engine* is responsible for managing the institutional reality by interpreting constitutive specifications. It encloses a *Constitutive_Program* component, responsible for storing and managing a constitutive specification. In a relation with the SAI model, the *Constitutive_Program* component incorporates the elements described in Section 3.2. Notice that the *Constitutive_Program* component is decoupled of specific constitutive specification languages and parsers. The elements of a constitutive program can be added by elements external to the *SAI_Platform* through the *IConstitutiveProgram* provided interface (Figure 17(a)). Parsers for the constitutive language proposed in Section 3.2.1 can be among these external elements. As the parsers are considered as external elements, different parser implementations can feed the *Constitutive_Program*. Even different constitutive languages, other than the one presented in Section 3.2.1, could be conceived and parsed to feed this same *Constitutive_Program*.

Besides the *Constitutive_Program*, the *Constitutive_Engine* contains also a *Constitutive_Reasoner*, that is a component responsible for interpreting constitutive specifications and for managing the constitutive state of the SAI. In a relation with the SAI model, the component *Constitutive_Reasoner* incorporates the elements described in Section 3.3. It is responsible, first, to keep a representation of the actual environment, that corresponds to the element X of the SAI state (cf. Definition 3.3.4). Based on this representation of the environment and on the constitutive program, the *Constitutive_Reasoner* is responsible to check the SFAs that must be created and dropped according to the stated in sections 3.3.2 and 3.3.3, building the constitutive state F as defined in Section 3.3.4.



(a) *IConstitutiveProgram* (b) *IEnvironmentalListener*



(c) *IConstitutiveListener*

Figura 17 – Interfaces of the *SAI_Platform*

The interface between the *SAI_Platform* and the elements external to the institution is done through two interfaces:

- The provided interface *IEnvironmentalListener* (Figure 17(b)) enables the environmental elements to input informations about their state into the *SAI_Platform*.
- The required interface *IConstitutiveListener* (Figure 17(c)) enables ex-

ternal components to be aware about the constitutive state. External components implementing this interface, that can be plugged in the *SAI_Platform*, are informed about SFAs that are added to and removed from the constitutive state.

Briefly, the dynamics involving the *IEnvironmentalListener* and the *IConstitutiveListener* evolves as follows: the environmental elements inform the *SAI_Platform* about changes in the environmental state. The *Constitutive_Reasoner* then follows the defined in sections 3.3.2 to 3.3.4 to check whether the new environmental state implies in changes in the constitutive state. If this is the case, the constitutive listeners are informed about the SFAs that have been added and removed. This dynamics is illustrated in the sequence diagram shown in Figure 18.

5.3 ADDING NORMS TO THE INSTITUTIONAL PLATFORM: THE *NORMATIVE_ENGINE*

The normative regulation in SAI may be provided by norms following different models, having thus, different implementations. For this reason, the proposed architecture envisages to enable to different normative engines to take part in the institution. To this end, the different normative engines must implement (or be embedded in some implementation of) the required interface *INormativeListener*. This interface is a specialization of *IConstitutiveListener* and, for this reason, the normative engines are informed about changes in the constitutive state when they occur. It is up to each normative engine to properly manage such information. In addition to the behaviour of a *IConstitutiveListener*, the components implementing *INormativeListener* have the operation *updateNormativeState*. The *Constitutive_Engine* triggers this operation when the a new constitutive state is achieved. The different normative engines are expected, through the operation *updateNormativeState*, to evaluate the normative state based on the new constitutive state. Thus, facing a new constitutive state, the normative engines check whether new normative states are also achieved. This dynamics is illustrated in the UML sequence diagram shown in Figure 18.

5.4 EXAMPLE OF IMPLEMENTATION

This section describes the insertion of the proposed architecture in an MAS infrastructure, being coupled with existing environmental and normative implementations. More precisely, it describes *SAI_Platform* as an insti-

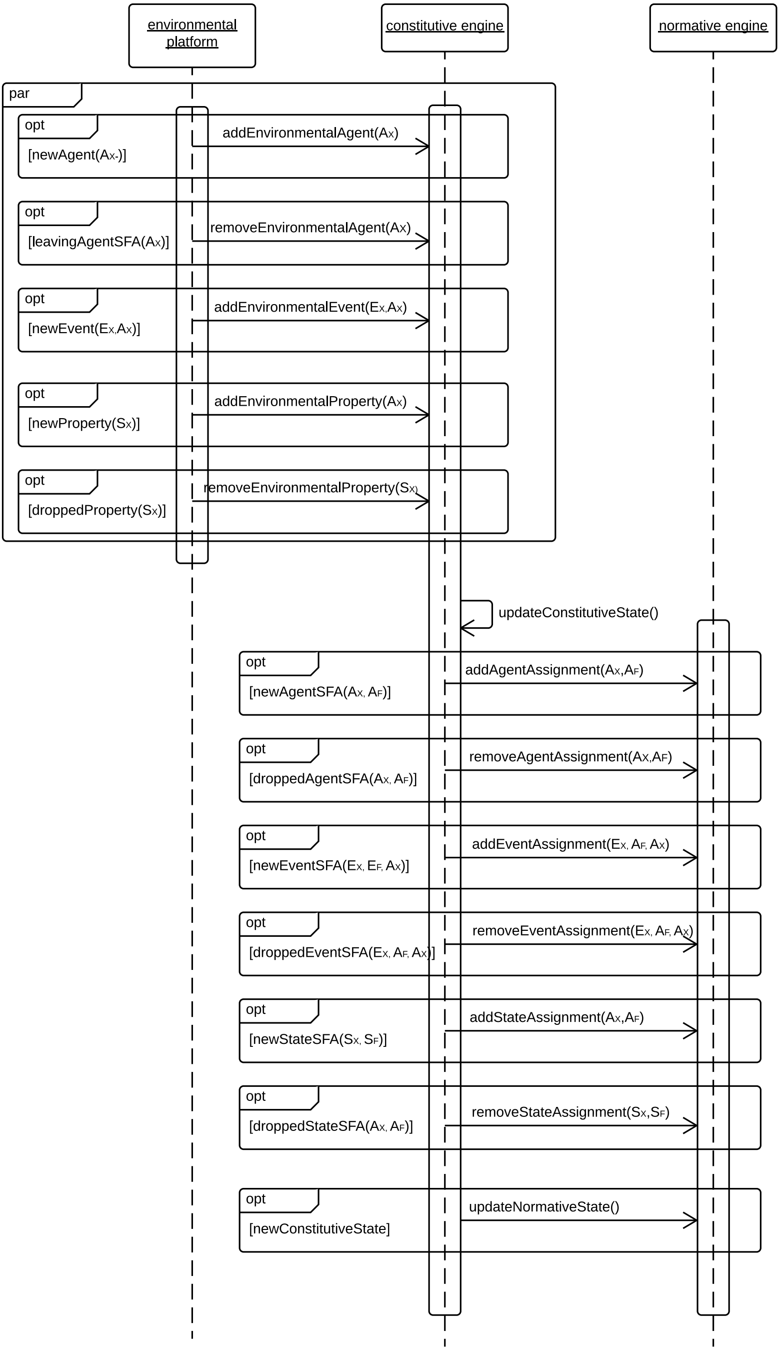


Figura 18 – Dynamics of the *SAI_Platform*

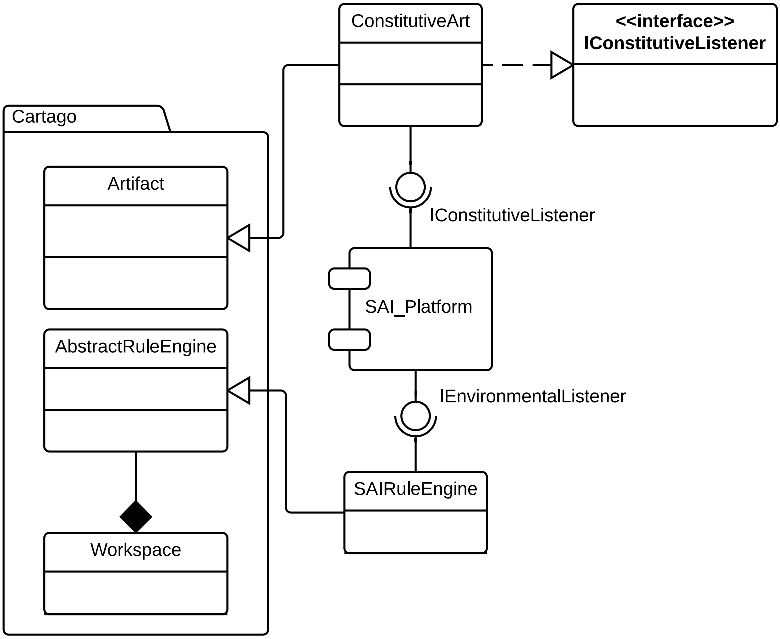


Figura 19 – Connecting the *SAI_Platform* with CArtAgO

tutional platform where (i) CArtAgO environments are the source of brute facts and (ii) the regulation is provided by a normative engine based on NPL norms.[1] The connection of the *SAI_ platform* with CArtAgO environments is described in Section 5.4.1. The insertion of a NPL engine in the *SAI_ platform* is described in Section 5.4.2.

5.4.1 Connecting the *SAI_ platform* to CArtAgO environments

The building blocks of CArtAgO[2] environments are the *artifacts*. They represent the different environmental elements that can be perceived and acted upon by the agents. The agents act upon the artifacts through available *operations* and perceive the state of the artifacts through their *observable properties*. Thus, in CArtAgO environments, the environmental events to be considered by SAI are produced when operations are performed in the

---

[1] An implementation of the *SAI_Platform* and of its interfaces with CArtAgO and NPL is available at `sitartinst.sf.net`.

[2] An implementation of CArtAgO is available at `cartago.sf.net`.

artifacts. The agents acting upon the artifacts are the environmental agents considered by SAI. The observable properties of the artifacts compose the environmental state to be considered by SAI.

CArtAgO environments are composed of one or more *workspaces*, that are logical places collecting the artifacts. Events occurring in the workspaces, triggered when operations are performed, as well as the changes in the observable properties, are caught by *rule engines*, that are specializations of the *AbstractRuleEngine* available in the CArtAgO machinery. In our proposed integration between SAI and CArtAgO, a rule engine called *SAIRuleEngine* is responsible to get the elements from the environment, inputting them in the *SAI_Platform*. In practical applications, it is possible to have several workspaces, each one with their corresponding *SAIRuleEngine* connected to the same *SAI_Platform*.

Having such connection, the *SAI_Platform* can compute the constitutive state based on the dynamics of CArtAgO environments. Informations about the current constitutive state are available to the agents as observable properties of the artifact *ConstitutiveArt*. When a new constitutive state is achieved, the *SAI_Platform* informs the *ConstitutiveArt*, as well as every other *IConstitutiveListener*, about the new constitutive state.

5.4.2 Inserting a NPL engine within the institution

As described in Section 5.3, normative engines are connected to the constitutive engine through the interface *INormativeListener* (cf Figure 16). To connect the NPL engine introduced in (HÜBNER; BOISSIER; BORDINI, 2011)[3] to the *SAI_Platform*, we conceive the class *Npl2Sai* (Figure 20). This class implements the behaviour of a *INormativeListener*, being thus informed about changes in the constitutive state and having also the method *updateNormativeState()* triggered by the constitutive engine.

Npl2Sai objects have access to the NPL machinery. The *NPLInterpreter* is responsible for interpreting NPL programs, changing the normative state based on a set of facts that, in this case, is the SAI constitutive state. These facts are stored in a *BeliefBase*. Thus, as soon as the *Npl2Sai* is informed about changes in the constitutive state, these changes are informed to this *BeliefBase*. The NPL engine has, thus, a consistent view about the current constitutive state to evaluate the norms. As described in Section 5.3, the *Constitutive_Engine* triggers the operation *updateNormativeState* in the normative listeners when the a new constitutive state is achieved. This method thus is triggered in the *Npl2Sai* objects connected to the constitutive engine.

---

[3] Available at `github.com/moise-lang/npl`.

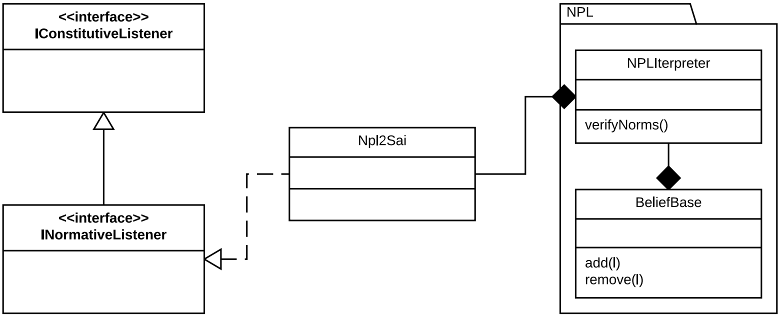


Figura 20 – The *Npl2Sai* class implements the behaviour of a *INormativeListener* and contains a *NPLInterpreter*

These objects then call the method *verifyNorms()*, provided by the *NPLInterpreter* make the norms to be evaluated with respect to the facts stored in the *BeliefBase*. This dynamics is shown in Figure 21.

5.5 CONCLUSION

This chapter proposes the architecture of an institutional platform to enable the deployment of institutions following the SAI model. Following the SAI conception, the proposed architecture provides the components for the platform both to manage the constitutive state and to incorporate different normative engines. Furthermore, the architecture is conceived to be accessible to heterogeneous environmental elements, even unknown in design time, to inform their state to the institutional platform.

These features can be seen in the described example of implementation, where the institutional platform is coupled with CArtAgO environments and has introduced a NPL normative engine. It is important to remark that the institutional platform is not conceived considering CArtAgO and NPL. Rather, it is conceived to enable the connections of environmental and normative elements without considering particular implementations. Having such connections requires, naturally some work of conceiving interfaces such as the *SAIRuleEngine* e *Npl2Sai*. But, in spite of these interfaces, the original implementations of CArtAgO and of the NPL Intepreter remain unchanged.

Some of the existing institutional platforms are conceived to manage an institutional reality that is limited to support a specific kind of norm. This is the case, for example, of the 2OPL (TESTERINK, 2012; DASTANI et al.,

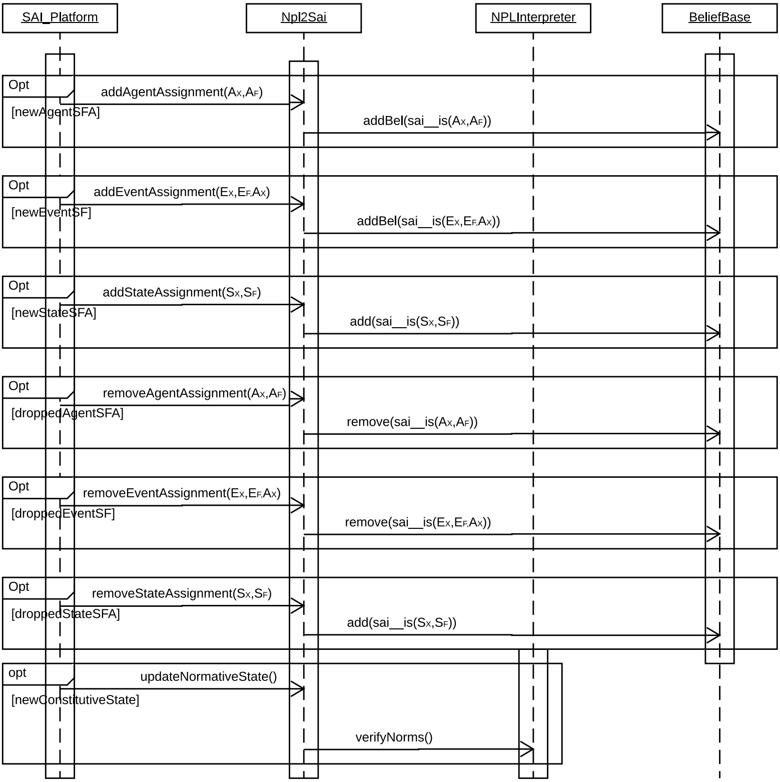


Figura 21 – Dynamics of the NPL interpreter inserted in the *SAI_Platform*

2009).[4] Our proposed architecture advances in this point as it is proposes an institutional platform where diferent normative engines, possibly managing different kinds of norms, can be inserted. The proposal of (BRITO; HÜBNER; BORDINI, 2013) also have a corresponding proposed infrastructure.[5] As described in Section 2.2.2.1, that model addresses the institutional reality as an interoperability issue, proposing ways to solve the gap between environmental and normative platforms. Our proposal disentangles conceptual and interoperability issues. The SAI model conceives the building blocks of artificial institutions independent of environmental, constitutive, or normative implementations. Interoperability issues are solved in the architectural level, as described in this chapter. The proposed architecture defines means input environmental elements into the institution, defining also means to the normative engines to be inserted within the institution.

---

[4] Available at `oopluu.sf.net`.

[5] Available at `countas.sf.net`.

6 APPLICATION EXAMPLE 1: CRISIS MANAGEMENT

Considering the problems and questions motivating this work, we deal with the hypothesis that Searle's theory is an inspiration to design artificial institutions where the institutional reality is connected (or connectable) both to norms and environment (cf. Section 1.3). This chapter presents an application example that helps us to evaluate whether and how SAI – that takes inspiration in Searle's theory to address the considered problems – confirms the proposed hypothesis. More precisely, we aim to evaluate the proposed approach for (i) representing the institutional reality, (ii) connecting such institutional reality to the environment, and (iii) basing the regulation of a given scenario. To this end, the roadmap to the experiment is (i) to design the institutional reality in terms of status functions, (ii) to design the connection of the institutional reality with the environment and (iii) to design the regulation based on such institutional reality. At the end, it is expected to evaluate whether the SAI provides means to (i) represent the institutional reality of the use case, (ii) to couple the institutonal reality with the environment, and (iii) to base the regulation of the scenario on the institutional reality.

The considered scenario is an extension of the crisis management scenario introduced in Section 3.2.1. In the following, Section 6.1 describes an use case related to the crisis management to be considered in this example; Section 6.2 describes the environment where the actors collaborate to manage the crisis; Section 6.3 describes the institutional specification (both constitutive and normative) to address the use case; Section 6.4 evaluates, based on the conceived application, whether SAI confirms the initial hipothesis. Besides checking whether the work on SAI confirms the initial hypothesis, this application example is useful to analyse what institutions as conceived by SAI add to MAS applications. Such discussion, built on the work described in (BRITO et al., 2015b, 2015c, 2015a, 2016b, 2016a), is presented in Section 6.5.

6.1 THE USE CASE

This application example considers a scenario where different, possibly distant actors (e.g. firefighters, police, citizens), collaborate to respond to disasters, within natural or artificial accidents, to limit material and human damages. In the context of crisis management, we consider a simplified but rich enough use case of crisis management where the goal is to deal with the evacuation of zones affected by a crisis. The actors, in this activity, are organised in three groups: a *Communal Command Post* (*CCP*) under the responsibility of the *Mayor*, a *Logistic Cell* (*LC*) and a *Support Cell* (*SC*), both controlled by the *CCP*, and the *Firefighters* (*FF*). Zones are classified as *secure* and *insecure*. The *Operational Command Post* (*OCP*), under the responsibility of the *Firefighters*, centralises and coordinates operational actions (i.e. fieldwork). The representative of the *Mayor* at the *OCP* is responsible for communication and coordination with the *CCP*.

There must be only one group of actors at a time to manage an evacuation. The *Mayor* is responsible for coordinating the evacuation of *secure* zones by commanding the *LC* and the *SC*. *FF* are the only responsible for evacuating *insecure* zones. When a zone is completely evacuated, the *SC* is responsible for registering the evacuated people. For simplicity, the only considered information provided by external data sources is the *phase* of the crisis in each zone: *preventive* (less severe) and *emergency* (more severe) – managed under dedicated policies.[1] When a crisis takes place, independent of its phase, the *Mayor* is obliged to designate his representative at the *OCP*. The named representative is responsible for establishing a permanent link between *OCP* and the the *Mayor* by regularly sending reports.

6.2 THE ENVIRONMENT: A COLLABORATION PLATFORM TO MANAGE CRISIS

This application example considers the platform proposed in (THÉVIN et al., 2015), conceived as an MAS, to support the collaboration of the possibly distant actors involved in a crisis management. Such platform is conceived as a network of TangiSense tables (KUBICKI; LEPREUX; KOLSKI, 2012) through which human actors interact (Figure 22). These tables can detect and locate tangible objects equipped with RFID tags. Their surface is further equipped with a liquid-crystal display (LCD) allowing a virtual display of complex simulations as well as virtual feedback connected to tangible objects.

In such a collaboration platform, the human actors place tangible objects on the tangible tables to signal their intended actions in the management of crisis. For example, an actor must place a specific object in a specific area of the table to signal the evacuation of a zone. Artificial agents also collaborate in the platform by performing tasks related to the infrastructure of

---

[1] It is not assumed that sensors, databases, GIS, etc. can evaluate and classify the phase of a crisis. The names *preventive* and *emergency* are used to make the example more illustrative. But in real scenarios, depending on the information source, these informations may have different identifiers.

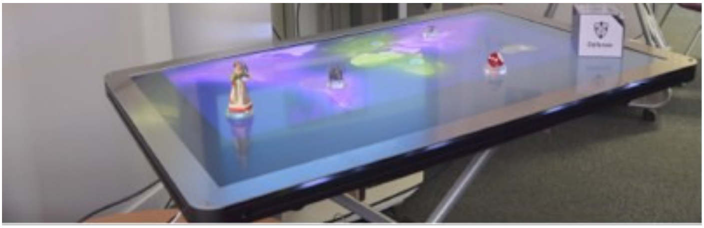


Figura 22 – TangiSense table (KOLSKI et al., 2015)

the system (e.g. delivering messages to the distant actors). Additional data sources, such as sensors, databases, geographic information systems (GIS), etc. are also part of the system, providing informations about the environmental variables related to the crisis (e.g. rainfall indexes, Richter Scale grades, etc.) (Figure 23).

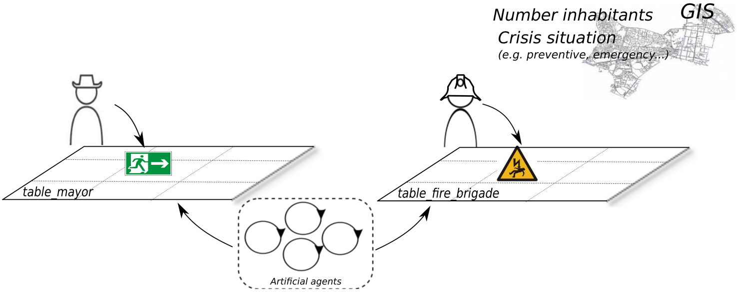


Figura 23 – Overview of the crisis management collaboration platform

6.2.1 The collaboration platform as the environment of the MAS

The network of tangible tables, as well as external data sources, compose the environment where the agents act. It is assumed that the human actors check in the tables before taking part in the crisis management. They use three kinds of tangible objects: *launch_tangible* to launch actions, *alert_tangible* to issue alerts, and *message_tangible* to send messages. Among all the observable events possibly occurring in the environment, the relevant ones here are (i) `checkin(AgentID, TableID)`, triggered when the agent `AgentID` checks into the table `TableID`, (ii) `putTangible(TableID, TangiID, X, Y, AgentID)`, triggered when the agent `AgentID` puts either a *launch_tangible* or an *alert_tangible* `TangiID` on the coordinates (X,Y) of `TableID`, and (iii) `putTangible(TableID, TangiID, X, Y, Target, Content, AgentID)`, triggered when the agent `AgentID` uses a *message_tangible* to send a message to texttTarget informing some `Content`.[2]

The relevant environmental properties that compose the environmental state, provided by databases, GIS, etc., are (i) `nbInhabitants(ZoneID,X)` holding when the `ZoneID` has `X` inhabitants and (ii) `security_phase(ZoneID, Phase)` holding when the `ZoneID` is on security `Phase` (s.t. `Phase` $\in \{preventive, emergency\}$).

6.3 SAI FOR CRISIS MANAGEMENT APPLICATION

The use case explained in Section 6.1 defines several regulative requirements that define how the different actors must behave. These requirements, however, are not attached to specific physical elements. Rather, the regulative requirements are defined considering the institutional reality, without specifying who are *mayor*, *firefighters*, etc., how *evacuations* are performed, what is a *secure* zone, and so on. For flexiblity facing a dynamic, complex environment, it is a requirement of the application to specify the norms using the concepts belonging to this institutional reality to keep the definition and management of norms independent of the physical world (FRANKE; CHAROY, 2010; ALDEWERELD et al., 2010). For example, norms specified in terms of *firefighters*, *evacuations*, etc. can range over different agents acting as firefighters and over different concrete actions employed to perform evacuations, that can even change along the systems' execution and be unknown in design time. The constitution of elements such as *mayor*, *firefighter*, *secure* and *insecure* zones, *evacuations*, etc. is up to the institution which norms are part of.

In the considered application, the use case is realised with (and the regulative requirements are applied to) the MAS deployed on top of a network of tangible tables that support the interactions among the agents. Since the acting of the actors in the tangible tables does not have per se any meaning in the crisis management, constitutive rules enable to *institutionalise* facts occurring in the environment, and to give them the proper meaning in the particular application (e.g. the tangible B in the position (C,D) counts as a command to evacuate the downtown). Such institutionalisation is important to the regulation of the scenario that is, ultimately, the regulation of the activities of the

---

[2] Elements in `true type` font appear in the SAI specification (Section 6.3.1.3). Terms starting with upper-case letters are variables.

actors in the environment. SAI provides such institutionalization when the environmental facts constitute status functions, building the institutional reality where the norms are based on. The next sections describe the institutional design for the considered use case: Section 6.3.1 describes the design of the institutional reality following the SAI approach while Section 6.3.2 describes the design of the regulation to be coupled with the institutional reality.

6.3.1 Crisis management SAI institutional reality

The institutional reality in SAI is represented by the constitutive state, that is composed of constituted status functions. Thus, to design the institutional reality in the considered use case, we define the status functions considered in the scenario, as described in Section 6.3.1.1, defining then how they are constituted (Section 6.3.1.2). Section 6.3.1.3 shows how these elements are expressed in a constitutive specification, that is illustrated in Figure 24.

6.3.1.1 Crisis Management SAI Status Functions

From the presented use case, it is possible to define the status functions that can be constituted to compose the institutional reality of the considered scenario.

- The **agent-status functions** define that agents act in the scenario as (i) *Mayor* of the town (`mayor`), (ii) member of the *LC* (`logistic_cell`), (iii) member of the *SC* (`support_cell`), (iv) `firefighter`, or (v) the representative of the *Mayor* in the *OCP* (`representative_ocp`).
- The **event-status functions** define that events occurring in the environment can mean in the institution (i) the command for an `Evacuator` to evacuate a `Zone` (`command_evacuation(Zone,Evacuator)`), (ii) the performance of an evacuation of a `Zone` (`evacuate(Zone)`), (iii) the support of the evacuation of a `Zone` (`support_evacuation(Zone)`, (iv) the registration of the evacuated people (`register_evacuated_people(Zone)`), and (v) the appointment of the representative of the *Mayor* at the *OCP* (`name_representative_ocp`).
- The **state-status functions** define that the system can be in states where, from the institutional perspective, (i) a `Zone` is considered secure for security procedures (`secure(Zone)`), (ii) a `Zone`

is insecure (`insecure(Zone)`), (iii) a `Zone` is electrical risky (`electrical_risky(Zone)`), and (iv) the expected time to send informations is expired (`max_time_to_inform`).

Notice that, although some status functions have names alluding to elements of the concrete world, such as "electric" and "time," they are not environmental elements. The status functions have these names to be more illustrative in this thesis. Naming status functions is part of the design of the institutional ontology and it is up to designer to properly choose the names.

6.3.1.2 Crisis Management SAI Constitutive Rules

The environmental dynamics described in the Section 6.2.1 animates the institutional dynamics when it gives rise to the constitution of the status functions. As for the status functions, three sets of constitutive rules are considered:

- **Agent-Status Function Constitutive Rules**. The rules 1 to 4 shown in the Figure 24 specify that the agent-status functions of `mayor`, `logistic_cell`, `support_cell`, and `firefighter` are constituted by the `Agent` that checks into the proper `Table` producing the event `checkin(Table,Agent)`. The `while` clause of the rule 1 still ensures that the status function of *Mayor* is assigned only to a single agent at a time as it defines that the agent keeps carrying such status function while it is not assigned to another agent or while it is assigned to the `Agent` itself. The rule 5 specifies that an agent counts as the representative of the *Mayor* at the *OCP* when it receives from the mayor a message whose content is `represent_mayor_ocp`.
- **Event-Status Function Constitutive Rules**. The rules 6 to 11 shown in the Section 6.3.1.3 define that some tangible interactions mean, in the institution, the command of an evacuation, the execution of an evacuation and the support to an evacuation. This meaning is conditioned to the tangible object used in the interaction and also to the *Actor* that performs the interaction. Besides defining the kind of action from institutional perspective, these constitutive rules define correspondences between different points of the tables and different geographic zones. As a result, we can consider that the coordinates (1,2) are related to the *downtown* while the coordinates (3,3) are related to the *industrial zone*.
- **State-Status Function Constitutive Rules**. By the

rule 14 shown in the Section 6.3.1.3, the property `security_phase(Zone,preventive)` holding in the environment counts as the `Zone` being secure for unprofessional people to deal with the security. By the first part of the `while` clause, such relation between environmental state (the zone being in preventive phase of crisis management) and institutional state (the zone being secure) holds while the zone does not pose electrical risks. Besides, by the remainder part of the `while` clause, such relation holds when the zone has, at most, 500 inhabitants or if it is already secure. Thus (i) if the property `security_phase(Zone,preventive)` starts to hold when the zone has more than 500 inhabitants, the zone is not considered secure and (ii) a zone remains secure even if its number of inhabitants changes exceeding the threshold. Notice that, if `security_phase(Zone,preventive)` does not hold in the environment, it cannot carry the status function `secure(Zone)`. The rules 15 and 16 define an `insecure(Zone)` from the institutional perspective. The rule 17 defines what constitutes an electrical risky zone. It specifies a *freestanding* assignment since there is not a concrete element in the environment to carry the status functions. The constitutive rule 18 specifies that the institution considers the `clock` showing a value multiple of 60000 as the deadline for reporting informations (the clock, in this case, is a counter incremented every millisecond).

6.3.1.3 Constitutive specification

From the previously described elements, it is possible write the constitutive specification, defining the conditions in which the elements composing the institutional reality are constituted. Such specification is shown below:

```
status_functions:
  agents: mayor, firefighter, logistic_cell, support_cell, representative_ocp.
  events: command_evacuation(Zone), evacuate(Zone), support_evacuation(Zone),
          register_evacuated_people(Zone), name_representative_ocp, link_mayor_ocp.
  states: secure(Zone), insecure(Zone), electric_risky(Zone), max_time_to_inform.
constitutive_rules:
                    /*Agent-Status Functions constitutive rules*/
  /*Actors carry the status functions according to their check in the tables*/
1: Agent count-as mayor
       when checkin(table_ccp,Agent) while not(Other is mayor)|Other==Agent.
2: Agent count-as logistic_cell
       when checkin(table_logistic_cell,Agent).
3: Agent count-as support_cell
       when checkin(table_support_cell,Agent).
4: Agent count-as firefighter when checkin(table_fire_brigade,Agent).
/* An actor is the mayor's representative at the OCP when the mayor
   sends it naming message */
5: Target count-as representant_ocp
     when putTangible(Table,message_tangible,X,Y,
```

```
                            Target,represent_mayor_ocp)[sai__agent(Actor)]
     while Actor is mayor.
               /* Event-Status Functions constitutive rules */
/* The mayor putting the object launch_tangible in the coordinates 1,2
   of any table counts as the command to evacuate the downtown */
6: putTangible(_,launch_tangible,1,2)[sai__agent(Actor)]
     count-as command_evacuation(downtown)
     while Actor is mayor.
/* Firefighter and logistic cell putting the object launch_tangible in the
    coordinates 1,2 of any table counts as the evacuation of the downtown */
7: putTangible(_,launch_tangible,1,2)[sai__agent(Actor)]
     count-as evacuate(downtown)
     while Actor is firefighter | Actor is logistic_cell.
/* The support cell putting the object launch_tangible in the coordinates 1,2
   of any table counts as supporting the evacuation of the downtown */
8: putTangible(_,launch_tangible,1,2)[sai__agent(Actor)]
     count-as support_evacuation(downtown)
     while Actor is support_cell.
/* Rules 9 to 11: similar to 6 to 8, but related to the industrial zone  */
9: putTangible(_,launch_tangible,3,3)[sai__agent(Actor)]
     count-as command_evacuation(industrial_zone)
     while Actor is mayor.
10: putTangible(_,launch_tangible,3,3)[sai__agent(Actor)]
     count-as evacuate(industrial_zone)
     while Actor is firefighter | Actor is logistic_cell.
11: putTangible(_,launch_tangible,3,3)[sai__agent(Actor)]
     count-as support_evacuation(industrial_zone)
     while Actor is support_cell.
/* The mayor names its representant at the OCP when it sends a message
   with the content "represent_mayor_ocp" */
12: putTangible(Table,message_tangible,X,Y,From,Target,represent_mayor_ocp)
                                                          [sai__agent(Actor)]
     count-as  name_representant_ocp
     while From is mayor.
/* The OCP is linked to mayor when the mayor's representant at
   OCP sends him a message with the content "crisis_report" */
13: putTangible(Table,message_tangible,X,Y,From,Target,crisis_report)
                                                       [sai__agent(Actor)]
     count-as link_mayor_ocp
     while From is representant_ocp & Target is mayor.
/* A zone preventive phase of crisis managament counts as that zone being
   secure if (i) it does not pose electrical risks and (ii) it has at most
   500 inhabitants */
14: security_phase(_,Zone,preventive) count-as secure(Zone)
     while not( AnyState is electric_risky(Zone)) &
         ((nbInhabit(_,Zone,X)& X<=500) |
          security_phase(_,Zone,preventive) is secure(Zone)).
/* A zone preventive phase of crisis managament counts as that zone
   being insecure if is electrical risky */
15: security_phase(_,Zone,preventive) count-as insecure(Zone)
        while AnyState is electric_risky(Zone).
/* A zone emergency phase of crisis managament allways counts as
   that zone being insecure */
16: security_phase(_,Zone,emergency) count-as insecure(Zone).
/* A zone is electrical risky if an actor counting as a firefighter puts
   the tangible alert_tangible in the coordinates (1,2) */
17: count-as electric_risky(downtown)
        when putTangible(_,alert_tangible,1,2)[sai__agent(Actor)]
        while Actor is firefighter.
/* The deadline to report informations is 60 seconds */
18: nticks(clock,Time) count-as max_time_to_inform
        while  (Time mod 60000==0).
```

Figure 24 – Constitutive specification for the crisis management scenario

6.3.2 Designing the regulation: norms for crisis management

The institutional reality produced from the elements described in Section 6.3.1 can base the different norms regulating the crisis management. In this application example, norms are expressed using an enhanced version of NPL, that besides obligations, enables us to express also prohibitions and permissions.[3]

The norms shown in Figure 25 define prohibitions and obligations to regulate to the use case described in Section 6.1. The norms do not refer directly to the environment. Rather, they refer to status functions. For example, the norm $n1$ specifies that, when the environment is in an state that counts as a `Zone` being secure, the agent carrying the status function of `mayor` is *obliged* to produce any event that means, in the institution, the command of an evacuation. *LC* and *SC* are obliged to react in different ways to this command (norms $n5$ and $n6$). Notice that the actions of *LC* and *SC* are triggered by the command of the *Mayor* independent of a zone being considered secure or insecure. This is why the *Mayor* is prohibited to command evacuation of insecure zones: to prevent *LC* and *SC*, that are non professional teams, to act when they are not expected to do so. *FF* are prohibited to evacuate secure zones (norm $n3$) but are obliged to evacuate insecure zones (norm $n4$). With this set of norms – $n1$ to $n6$ – we clearly define the expected coordinated behaviour of the different actors with respect to the evacuation activities. The norm $n7$ defines that, after a zone being evacuated by any actor, the *SC* is obliged to register the evacuated people. The norm $n8$ specifies that, if there is at least one zone in any phase of crisis, then the *Mayor* must name their representative at *OCP*. By the norm $n9$, this representative is always obliged to keep a link between the *Mayor* and the *OCP*.

6.4 EVALUATING SAI AGAINST THE INITIAL HYPOTHESIS

This section evaluates whether and how SAI provides means (i) to represent the institutional reality of the use case, (ii) to connect the institutonal reality to the environment, and (iii) to base the regulation of the scenario in the institutional reality. These points are discussed below.

- **Representation of the institutional reality.** The notion of institutional reality is implicit in the use case described in Section 6.1 as it describes how the actors must behave without referring to the concrete environmental elements related to such behaviour. SAI, through the

---

[3] Available at `github.com/moise-lang/npl`.

```
/* The mayor is obliged to command evacuations of secure zones */
norm n1: secure(Zone)
   -> obligation(mayor,n1,command_evacuation(Zone),‘now’+‘1 day’).
/* The mayor is prohibited to command evacuations of insecure zones  */
norm n2: insecure(Zone)
   -> prohibition(mayor,n2,command_evacuation(Zone),‘now’+‘1 day’).
/* The firefighter is prohibited to evacuate secure zones */
norm n3: secure(Zone)
   -> prohibition(firefighter,n3,evacuate(Zone),‘now’+‘1 day’).
/*  The firefighter is obliged to evacuate insecure zones */
norm n4: insecure(Zone)
   -> obligation(firefighter,n4,evacuate(Zone),‘now’+‘1 day’).
/* The logistic cell is obliged to evacuate a zone when a command is emitted */
norm n5: command_evacuation(Zone)
   -> obligation(logistic_cell,n5,evacuate(Zone),‘now’+‘1 day’).
/* The support cell is obliged to support the evacuation of a zone when a command is emitted */
norm n6: command_evacuation(Zone)
   -> obligation(support_cell,n6,support_evacuation(Zone),‘now’+‘1 day’).
/* The support cell is obliged to register evacuated people */
norm n7: evacuate(Zone)
   -> obligation(support_cell,n7,register_evacuated_people(Zone),‘now’+‘1 day’).
/* Mayor must name their representant on OCP when there is at least one zone in
   any phase of crisis */
norm n8: secure(Zone) | insecure(Zone)
   -> obligation(mayor,n8,name_representant_pco ,‘now’+‘1 day’).
/* The mayor’s representant in OCP is always obliged to keeps a link between mayor and OCP */
norm n9: true
   -> obligation(representant_ocp,n9,link_mayor_ocp,‘now’+‘1 day’).
```

Figura 25 – Norms for crisis management

status functions, provides means to make such institutional reality explicit. The components of the institutional reality implicit in the described use case are explicitly designed in terms of the agent-, event-, and state-status functions defined in Section 6.3.1.1.

- **Connection between institutional reality and environment.** The inspiration in the Searle’s theory makes the institutional reality, in SAI, to be not merely *connected*, but *constituted* from the environment. The connection between the institutional reality and the environment resides precisely in this aspect: the components of the institutional reality are counterparts of the environmental elements. The constitutive rules described in Section 6.3.1.2, backed by the semantics defined in Chapter 3, define the conditions under which the the status functions are constituted whithin the institutional reality.

- **Basing regulation on the institutional reality.** By the Searle’s theory, the *deontic powers* of the individuals are attached to status functions. SAI provides such attachement inasmuch as norms, that represent the deontic powers, are specified in terms of status functions. The “world” addressed by the norms is the institutional reality, that is composed of status function assignments. The way this “world” is addressed is discussed in Chapter 4. As well as the regulative requirements defined in the use case, the specified norms do not refer to concrete environmental elements.

It is possible to see, thus, that SAI provides the elements to represent artificial institutions where the institutional reality is constituted from the environment to support the normative regulation. An important aspect to be observed is that SAI provides a clear separation between environment, institutional reality, and norms. Norms specify the regulative requirements without any concern about to represent or to manage the institutional reality they refer to. Status functions and constitutive rules define the institutional reality without any concern about to represent or to manage the normative regulation. The environmental elements constitute the institutional reality that affect the normative regulation without any concern about to represent and manage institutional elements.

6.5 CONTRIBUTIONS OF THE SAI APPROACH TO COMPLEX CRISIS MANAGEMENT ISSUES

This section discusses, based on the designed application, what SAI brings to the system when it allows to specify constitution and regulation as independent but connected dimensions of the institution. The discussions are based on the constitutive specification illustrated in Section 6.3.1.3 and on the norms illustrated in Figure 25, unless stated otherwise.

6.5.1 Keeping independent normative and constitutive layers

Having norms based on the abstraction provided by the constituted status functions allows to specify the regulation considering the crisis domain independent of – but still connected to – the environment where the crisis management takes place. With this clear separation between normative and constitutive levels, the constitutive rules may change without requiring to change the norms. This is an advantage as the norms can remain stable even when their relation with the environmental elements changes. For example, the way the *Mayor* commands evacuations could change from a tangible action to the sending of a message to the *LC*. To introduce this change in the scenario, we can replace the constitutive rule 6 by the one shown below, without any change in the norms related to the command of evacuation (1 and 2):

```
6: send_message(From,To,‘‘evacuate downtown’’)[sai__agent(Actor)]
     count-as command_evacuation(downtown)
     while From is mayor & To is logistic_cell
```

The contrary is also possible: the regulative requirements can change even when the institutional reality remains unchanged. That is to say, norms can be changed without changing the constitutive rules. Consider for example the norm 2. It states that the `mayor` is prohibited to command evacuations of a `Zone` when it is `insecure`, i.e. when the zone is in emergency phase of crisis or it poses some electrical risk or it has more than 500 inhabitants (constitutive rules 8 to 10). The regulative requirements could evolve to consider electrical risk as the only condition prohibiting the mayor to command the evacuation. To reflect this evolution, the constitutive rules could remain as they are and the norm $n1$ can be changed to:

```
norm n1: electric_risky(Zone)
   -> prohibition(mayor,n1,command_evacuation(Zone),‘now’+‘1 day’).
```

6.5.2 Avoiding discrepancies in the interpretation of the environment

When the regulation is expressed through abstract concepts, the different actors collaborating in the crisis management may have different particular interpretations about the same fact occurring in the environment. To be effective, however, their efforts must be coordinated based on the same interpretation about each situation (i.e. based on the interpretation provided by the institution “crisis management”). Consider, for example, that, for the *Mayor*, a zone is *secure* whenever it is in preventive phase and its number of inhabitants is below a certain threshold. For the *FF*, conversely, a zone is *secure* whenever it is in preventive phase and posing no risk, such as an electrical one. That is to say, a *secure* zone is differently constituted in *Mayor* and *FF* particular perspectives. This may lead to incoherences in the regulation of the actions as the same action can be considered as mandatory and forbidden according to the perspective of the different actors. For example, if the downtown is on secure phase of crisis management, has 1000 inhabitants, and is not electrical risky, then *FF* consider themselves as prohibited to evacuate it as it is a *secure* zone (norm 3). For the *FF*, such evacuation is up to the *Mayor* as he is the responsible to evacuate secure zones (norm 1). The *Mayor*, on its turn, does not consider itself as responsible to evacuate the downtown as it does not consider it as a *secure* zone (i.e. for the *Mayor*, the norm 1 is not activated). Thus, neither the mayor nor the firefighters consider themselves as responsible to evacuate an endangered zone.

SAI helps to avoid these incoherences, as it conceives the institution specifying not just the norms but also the grounding in the environment of the concepts used in the norms. Making relations between the abstract concepts used in the norms and the environmental elements is an institutional task instead of being up to the agents. In the aforementioned case, the constitutive rule 14 expresses the *institutional* conception of a *secure* zone, independent of the particular view from the actors about what a secure zone is.

Notice that conflicts in the interpretations may arise in real time, being thus unpredictable in design time. These cases, naturally, cannot be solved by specifying constitutive rules in design time. In this case, empowered agents (humans or artificial) could define new constitutive rules at run time.

6.5.3 Contextualising the set of active norms

The expected actors' behaviour may be different under different contexts. Situating the normative regulation following the SAI approach enables us to have such a contextualisation considering the different contexts of the environment where the collaboration takes place. This section presents two examples of such a contextualisation. Section 6.5.3.1 shows the active norms evolving according to the evolution of the phase of the crisis. Section 6.5.3.2 shows the set of active norms being defined according to the zone under security procedures.

6.5.3.1 Contextualising norms according to the phase of the crisis

Norms can evolve automatically, depending on the phase of the crisis. As already mentioned, in preventive phase, the *Mayor* is obliged to perform the evacuation of zones whose number of inhabitants is lower than a threshold. When the phase changes to emergency, the *Mayor* becomes forbidden to perform evacuations and it is mandatory for the *FF* to do it. In preventive phase, the environmental property `security_phase(Zone, preventive)` always holds. If `Zone` is not electrical risky and has at most 500 inhabitants, then the status-function `secure(Zone)` is constituted by the constitutive rule 14. As a consequence, the norms 1 and 4 become active. When moving to emergency phase, the previous environmental property is modified to `security_phase(Zone, emergency)` and the status function `secure(Zone)` is not constituted anymore. The status function `insecure(Zone)` is now constituted according to the constitutive rule 16, which activates norms 2 and 4. As may be seen, by changing the context (from *preventive* to *emergency*), even if the environmental facts are interpreted with the same set of constitutive rules, the set of active norms will change.

6.5.3.2 Spatial contextualisation of norms

The constitution of secure and insecure zones evolves particularly according to each zone. Different zones may be in different phases of the crisis, may have different classifications regarding to electrical risks, may have different number of inhabitants, etc. These differences require particular sets of active norms according to the zones. Basing the norms on the interpretation of the environment provided by the constitutive rules enables such spatial contextualisation of the active norms. For example, if the GIS informs the properties `security_phase(downtown, preventive)` and `security_phase(industrial_zone, emergency)`, then the *downtown* is considered secure while the *industrial zone* is considered insecure according to the constitutive rules 14 and 16. As consequence, the set of obligations and prohibitions standing for the different actors is different in the different zones.

6.5.4 Contextualising the normative lifecycle

Norms, even unchanged, can be activated, violated, and fulfilled under different environmental conditions depending on the context. For example, in certain circumstances, the *SC* is obliged to support the *LC* on evacuation procedures. By the constitutive rule 11, such obligation is fulfilled in the collaboration platform when the actor carrying the status function of `support_cell` puts a `launch_object` on the proper coordinates of the table. But we could imagine more contextualised norms such that the actions performed by the *SC* to fulfil the obligation to `support_evacuation` are different according to the evacuated zone. For example, using a `launch_object` could mean the support for the evacuation of the downtown while sending a weather report to the *LC* could mean the support for the evacuation of the industrial zone. The constitutive rule 11 could be split in two rules as follows:

```
11a: putTangible(_,launch_tangible,1,2)[sai__agent(Actor)]
         count-as support_evacuation(downtown)
         while Actor is support_cell.

11b: putTangible(_,message_tangible,3,3,Target,
                               weather_report)[sai__agent(Actor)]
         count-as support_evacuation(industrial_zone)
         while Actor is support_cell & Target is logistic_cell.
```

This example illustrates the contextualisation of the norm fulfilments. But the same idea applies to norm activations, violations, etc. For example, the norms 5 and 6 are activated when an evacuation command is emitted. By the constitutive rule 6, such a command is constituted by the event of the `mayor` putting a `launch_object` on the table. This constitution could be contextualised so that, for example, in certain circumstances, the command of evacuation is constituted by either (i) actions other than putting a a `launch_object` or (ii) the same action, triggered by actors others than the *Mayor*.

6.5.5 Assigning norms independent of the actions of the assignees

In SAI, the expected agents' behaviour in the society is conditioned by the agent-status functions that they carry. Sometimes the agents have some control over their standing obligations and prohibitions because they have some control over the agent-status function that they carry. But the assignment of agent-status functions may be conditioned by circumstances that are beyond the control of the assignees and, in this case, the agents do not have control over their standing obligations and prohibitions. These circumstances may be even intentionally produced by an agent that intend to create obligations to another one (kind of normative transmission process (HOLLANDER; WU, 2011)).

For example, actors do not have obligations and prohibitions targeted to *FF* unless they actively check in the *table_fire_brigade* (constitutive rule 4). But an actor becomes *representative_ocp* – having thus new obligations – exclusively due to an action performed by the *Mayor* (constitutive rule 5).

6.5.6 Designing empowerment

The notion of constitution has been employed to model the institutionalised power, i.e. the power of agents to produce facts in the institution by the performance of specific kinds of actions in certain conditions (JONES; SERGOT, 1996). The conditions of empowerment usually include the position occupied by the agent in the institution, that, in SAI, is captured by the agent-status functions. SAI allows us to design the institutionalised power in the crisis scenario. For example, by the constitutive rule 12, an actor carrying the status function of *Mayor* is the only one having the power, by sending a message, to make another one as the representative of the *Mayor* at *OCP*. The same action, performed by an agent that does not carry the status function of *Mayor*, does not have such effect.

6.6 CONCLUSIONS

As evaluated in this chapter, the elements borrowed from Searle's theory to compose artificial institutions – status functions, constitutive rules, and norms – enable us to design artificial institutions where the institutional reality is constituted from the environment and where the normative regulation is based on. This confirms the initial hypothesis of Searle's theory being an inspiration to design artificial institutions where it is possible to represent the institutional reality, connecting it to the environment and to the norms.

Having means to design institutions as conceived by SAI – having explicit, decoupled, but still connected representations of institutional reality and norms – adds some power to MAS development. In the particular application example, SAI helps to meet two requirements of designing a tool for crisis management: clear coordination and flexibility, that are necessary but not trivial in the conceived platform (FRANKE; CHAROY, 2010). Flexibility requires to have norms decoupled of the environment, being thus specified in abstract terms. Clear coordination requires to provide an unified interpretation of these abstract terms with respect to the environment. These requirements clearly point to the two dimensions of institutional design considered in SAI: constitution and regulation. SAI provides to the application the means to specify these two dimensions in an independent but well connected way. That is to say: it is possible to specify constitution independent of what are the regulative requirements expressed by the norms and it is also possible to specify norms independent of how the status functions are constituted.

7 APPLICATION EXAMPLE 2: BUILD A HOUSE

This chapter describes a second application example, where SAI is applied to the *build-a-house* example, that is an MAS originally conceived to illustrate multi-agent programming using the JaCaMo framework (BOISSIER et al., 2013). The original implementation of the *build-a-house* includes the notion of regulation but does not include notion of *institution*.[1] Norms are designed using abstract concepts that are not related to the environment. But these abstract concepts are not part of the institutional reality and, thus, are not constituted from the environment. The regulation, thus, is not situated.

In this application example, the original implementation is extended so that it has introduced the notion of institution as conceived by SAI. The abstract concepts used in the norms become components of the institutional reality, that, on its turn, is constituted from the environment where the agents act. It is possible, thus, to analyse the employment of SAI to explicitly design institutional elements that are originally implicit in an existing application. Furthermore, having two versions of the same MAS – the original one without the notion of institution, and the extended one including SAI – enables us analyse what is added to the MAS by introducing SAI. Finally, it is possible to compare the proposed model of institutional reality with another one that has been applied to this same *build-a-house* example in (BRITO; HÜBNER; BORDINI, 2013).

In the following, Section 7.1 describes the normative model and platform used in the *build-a-house* example; Section 7.2 describes how the regulation provided by that model is related to the institutional reality as conceived by SAI; Section 7.3 describes the *build-a-house* example, starting by the original version and then moving to a new version where the regulation is part of SAI, discussing also the relevant aspects of including SAI in the system.

7.1 REGULATION IN JACAMO: NOPL AND ORA4MAS

The regulative requirements, in JaCaMo, are modelled as $\mathcal{M}$OISE organisations that are, in runtime, translated to norms following the Normative Organisation Programming Language (NOPL) (HÜBNER; BOISSIER; BORDINI, 2011). NOPL is a particular class of NPL (with same syntax and semantics) specialised for expressing, through norms, the regulative requirements of $\mathcal{M}$OISE organisations.

---

[1] The notions of *institution* and *institutional reality* as taken in this thesis are not part of the current JaCaMo model.

Besides norms, a NOPL program contains static facts and rules to express various static concepts and properties of a $\mathcal{M}$OISE specification (e.g. cardinalities of missions and groups, relations between roles and missions, etc.). The static facts and rules do not change in runtime unless the organisational specification also changes. On the other hand, the NOPL normative dynamics is animated by *dynamic facts* that depend on runtime dynamics external to the NPL engine. The possible dynamic facts defined in (HÜBNER; BOISSIER; BORDINI, 2011) are:

- $play(a,\rho,gr)$: the agent a plays a role ρ in the group gr
- $responsible(gr,s)$: the group instance gr is responsible for the scheme s
- $comitted(a,m,s)$: the agent a is committed to the mission m in the scheme s
- $achieved(s,g,a)$: the goal g in scheme s has been achieved by the agent a

It is important to remark that the organisational semantics of the dynamic facts does not belong to NOPL semantics. Concepts such as *role*, *goal*, etc., are not part of the language and $play(a,\rho,gr)$, $achieved(s,g,a)$, etc. are just predicates used to write the norms.

In JaCaMo, the NOPL programs are embedded in two kinds of ORA4MAS artifacts: *GroupBoard* and *SchemeBoard* (figures 26 and 27) (HÜBNER et al., 2009; HÜBNER; BOISSIER; BORDINI, 2011). The dynamic facts are produced when the agents execute some available operations in the artifacts. For example, the execution of the operation *commit_mission* in a *SchemeBoard* produces the dynamic fact $comitted(a,m,s)$.

Figure 28 illustrates two norms of a NOPL program. The norm $n1$ defines that, under certain conditions, an agent A is obliged to produce the dynamic fact $committed(A,management\_of\_house\_building,S)$. The norm *ngoal* defines that, under certain conditions, an agent A is obliged to produce the dynamic fact $achieved(S,G,A)$. Related to the Moise semantics, $n1$ expresses that under certain conditions, the agent A is obliged to commit to the mission $management\_of\_house\_building$ while the $n2$ expresses that the agent A is obliged to achieve the goal G.

7.2 SITUATING NOPL

In SAI approach, an institution is situated when the dynamics of the normative state is based on the constitutive one. As the facts that animate the

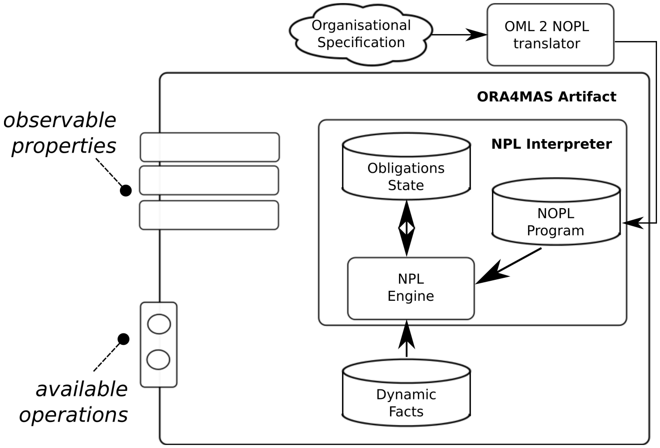


Figura 26 – General schema of an ORA4MAS artifact (adapted from (HÜBNER; BOISSIER; BORDINI, 2011))

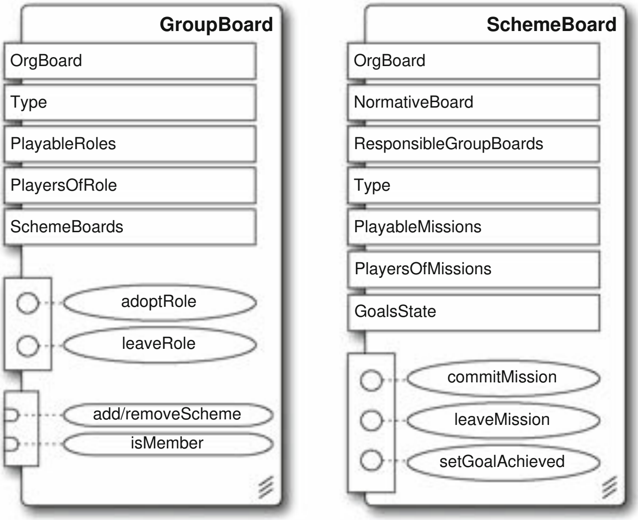


Figura 27 – ORA4MAS artifacts (HÜBNER et al., 2009)

```
norm n1:
        scheme_id(S) & responsible(Gr,S) &
        mplayers(management_of_house_building,S,V) & V < 1 &
        fplay(A,house_owner,Gr) &
        not mission_accomplished(S,management_of_house_building)
     -> obligation(A,n1,committed(A,management_of_house_building,S), ‘now‘+‘2 minutes‘).

norm ngoal:
        committed(A,M,S) & mission_goal(M,G) & goal(_,G,_,achievement,_,D) &
        well_formed(S) & not satisfied(S,G) & enabled(S,G) &
        not super_satisfied(S,G)
     -> obligation(A,ngoal(S,M,G),achieved(S,G,A),‘now‘ + D).
```

Figura 28 – Example of NOPL norms

NOPL dynamics are the *dynamic facts*, then situating the regulation provided by NOPL requires that the dynamic facts are part of the constitutive state, i.e. they must be *constituted* from the environmental elements.

The dynamic facts can be seen as states, i.e. they represent the state of agents playing roles, groups being responsible for schemes, agents being committed to missions, and goals being achieved. In SAI, the dynamic facts can be viewed as state-status function assignments, as explained below:

- $play(A,R,G)$. There are environmental and constitutive states that count as the agent A playing the role R in the group G.

- $responsible(G,S)$. There are environmental and constitutive states that count as the group G being responsible to execute the scheme S.

- $committed(A,M,S)$. There are environmental and constitutive states that count as the agent A being committed to the mission M in the scheme S.

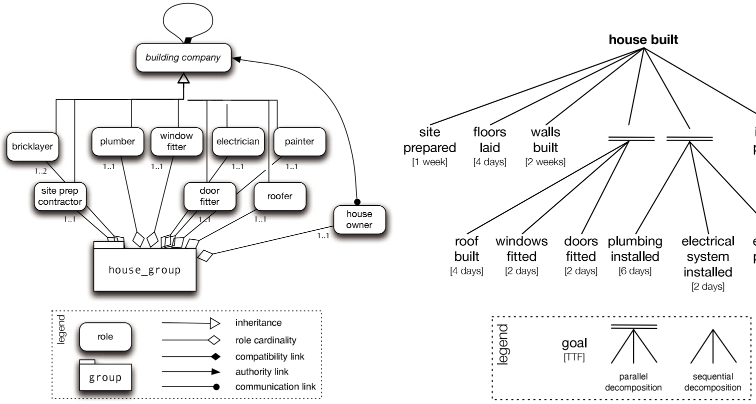
- $achieved(S,G,A)$: There are environmental and constitutive states that count as the goal G of the scheme S has been achieved by the agent A.

7.3 THE *BUILD-A-HOUSE* EXAMPLE

The original implementation of the *build-a-house* example is described in Section 7.3.1. The new implementation, where SAI is introduced, is described in Section 7.3.2. Discussions and comparisons about involving the new implementation are found in Section 7.3.3.

7.3.1 The original implementation

The *build-a-house* is an MAS representing the inter-organisational workflow involved in the construction of a house. To build a house, specialised companies are hired when they win auctions related to the different required tasks (e.g. to prepare the site, to build the walls, etc.). Once hired, the companies must work in a coordinate manner to build the house. For example, the walls must be built before the building of the roof, such a building must be done by a bricklayer, etc. Such coordination is guided by a $\mathcal{M}$OISE organisation where the the agents play some roles and, by doing so, become responsible for some goals in the building of the house. The organisational specification is shown in Figure 29 using the $\mathcal{M}$OISE notation. The organizational specification is translated to the NOPL programs shown in Appendix A.



norm	role	mission / goals
n1	house owner	house built
n2	site prep contractor	site prepared
n3	bricklayer	floors laid, walls built
...		

Figura 29 – Organisational specification of example *Build-a-House*: structural specification (left), functional specification (right) and normative specification (bottom) (BOISSIER et al., 2013).

The original example is not situated, i.e. there is not a connection between what happens in the environment and the normative state. The agents are responsible to interpret the environmental facts informing, through the ORA4MAS artifacts, that they are playing roles, committed to missions, responsible for schemes, and that goals have been achieved.

7.3.2 *Build-a-house* with SAI

A new version of the MAS has been implemented where the environmental dynamics produces constitutive states where the dynamic facts are constituted as state-status function assignments. We develop the constitutive specification whose excerpt is illustrated in Figure 30. The complete specification is in the Appendix B. The constitution of the state-status functions representing the dynamic facts considered in NOPL is explained below:

- $play(A,R,G)$. The constitutive rule 1 defines that state-status function *play(giacomo,house_owner,"hsh_group")* is constituted under any condition as (i) it is a *freestanding assignment*, i.e. there is not a state that must hold to carry the state-status function and (ii) the *when* and *while* clauses are omitted in the constitutive rule. In the *build-a-house* scenario, it means that, in any circumstance, the agent *giacomo* plays the role *house_owner*. The constitutive rule 2 defines that the state-status function *play(Agent,site_prep_contractor,"hsh_group")* is constituted if the property $currentWinner(auction\_for\_SitePreparation,Agent)$ holds in the environment when 5000 milliseconds have been elapsed. In the *build-a-house* scenario, it means that, in the specified circumstances, $Agent$ is playing the role $site\_prep\_contractor$. The constitutive rule 3 is similar considering the auction $auction\_for\_Floors$ and the role $bricklayer$. There are other similar constitutive rules, omitted in chapter but detailed in Appendix B, defining that the winners of the other auctions play the roles illustrated in the Figure 29.
- $responsible(G,S)$. The constitutive rule 11 defines that the state-status function *responsible("hsh_group","bhsch")* is constituted when a set of assignments of the state-status function $play(A,R,G)$ is also constituted. In the *build-a-house* scenario, it means that the group *hsh_group* is responsible for executing the activities prescribed by the scheme *bhsch* when a specific number of agents is playing each role.
- $committed(A,M,S)$. The constitutive rule 12 defines that the state-status function *committed(A,management_of_house_building,"bhsch")* is constituted when the state-status function *play(A,house_owner,"hsh_group")* is also constituted. In the *build-a-house* scenario, it means that that an agent playing the role *house_owner* counts as this agent being committed to the mission $management\_of\_house\_building$. The constitutive rule 13 is similar considering the role $site\_prep\_contractor$ and the mission $prepare\_site$. There are other 8 similar constitutive rules, omitted in

this chapter but detailed in Appendix B, defining the conditions that count as the agents being committed to the missions illustrated in the Figure 29.

- $achieved(S,G,A)$: The constitutive rule 23 defines that the state-status function *achieved("bhsch",site_prepared,A)* is constituted when the event *prepareSite[sai__agent(A)]* occurs in the environment. In the *build-a-house* scenario, it means that the goal $site\_prepared$ is achieved by the agent A when A produces, in the environment, the event $prepareSite$. This is a freestanding assignment as there is not a state carrying the status function *achieved("bhsch",site_prepared,A)*. The constitutive rule 24 is similar, defining that an agent achieves the goal *electrical_system_installed* when it produces, in the environment, the event *installElectricalSystem*. There are other similar constitutive rules, omitted in this chapter but detailed in Appendix B, defining the achievement of the goals illustrated in the Figure 29 when some environmental events occur.

These constitutive rules define the constitution of all the dynamic facts required to animate the dynamics the of NOPL programs generated from the organisational specification of the *build-a-house* example.

7.3.3 Discussion of the example

Having developed a new version of the *build-a-house* example, it is possible to discuss some relevant aspects of introducing SAI in this MAS. Section 7.3.3.1 discusses some conceptual aspects of contextualizing NOPL in an institution. Section 7.3.3.2 analyses what SAI adds to the original implementation. Section 7.3.3.3 compares the situatedness provided by SAI to the regulation to the one provided by a different approach to the same application.

7.3.3.1 Institutionalisation of the norms

As stated in the introduction of this chapter, a first point to analyse in this application example is the introduction of SAI in an MAS that contains norms but that is originally developed without considering an explicit notion of institution. The normative model regulating the *build-a-house* example is the NOPL, that is a specialisation, having the same syntax and semantics, of the NPL. Thus, the coupling of the NOPL representations and dynamics with

```
 status_functions:
  states: play(A,R,G), responsible(G,S), committed(A,Mission,S),
          achieved(S,G,A), well_formed(Group).

constitutive_rules:
  1:  count-as play(giacomo,house_owner,"hsh_group").

  2: currentWinner(auction_for_SitePreparation,Agent)
      count-as play(Agent,site_prep_contractor,"hsh_group")
      while nticks(clock,Time)&(Time>5000).

  3: currentWinner(auction_for_Floors,Agent)
      count-as play(Agent,bricklayer,"hsh_group")
      while nticks(clock,Time)&(Time>5000).

     ...

  11:  count-as responsible("hsh_group","bhsch")
       while play(Electrician1,electrician,"hsh_group")  &
              not(play(Electrician2,electrician,"hsh_group") &
                   not(Electrician1==Electrician2)) &
             play(Site_prep_contractor1,site_prep_contractor,"hsh_group") &
              not(play(Site_prep_contractor2,site_prep_contractor,"hsh_group") &
                   not(Site_prep_contractor1==Site_prep_contractor2))&
             play(Bricklayer1,bricklayer,"hsh_group") &
             play(Bricklayer2,bricklayer,"hsh_group") &
             not(play(Bricklayer3,bricklayer,"hsh_group") &
                   not(Bricklayer1==Bricklayer2) & not(Bricklayer2==Bricklayer3))&
             play(Plumber1,plumber,"hsh_group")  &
              not(play(Plumber2,plumber,"hsh_group") &
                   not(Plumber1==Plumber2)) &
             play(Window_fitter1,window_fitter,"hsh_group") &
              not(play(Window_fitter2,window_fitter,"hsh_group")&
                   not(Window_fitter1==Window_fitter2))&
             play(Door_fitter1,door_fitter,"hsh_group")&
              not(play(Door_fitter2,door_fitter,"hsh_group") &
                   not(Door_fitter1==Door_fitter2)) &
             play(Roofer1,roofer,"hsh_group") &
              not(play(Roofer2,roofer,"hsh_group") & not(Roofer1==Roofer2))&
             play(House_owner1,house_owner,"hsh_group") &
              not(play(House_owner2,house_owner,"hsh_group") &
                   not(House_owner1==House_owner2)).


  12: play(A,house_owner,"hsh_group")
       count-as committed(A,management_of_house_building,"bhsch").

  13: play(A,site_prep_contractor,"hsh_group")
       count-as committed(A,prepare_site,"bhsch").

       ...

  23:  count-as achieved("bhsch",site_prepared,A)
       when prepareSite[sai__agent(A)].

  24:  count-as achieved("bhsch",electrical_system_installed,A)
       when installElectricalSystem[sai__agent(A)].
```

Figura 30 – Excerpt of the constitutive specification of the *build-a-house* example

SAI is, in fact, the coupling of the NPL, described in Section 4.2.2.

As the coupling between NPL and SAI is discussed in Chapter 4, we can focus on analysing the introduction of SAI in the *build-a-house* from the perspective of the design of the application. The state of a NOPL program evolves based on the occurrence of the dynamic facts described in Section 7.1. Neither the source of these dynamic facts nor their semantics are defined by the NPL/NOPL model. Introducing SAI provides a well defined source of the dynamic facts: the institutional reality represented by the constitutive state. Furthermore, the dynamic facts get a well defined semantics: they are SFA.

The dynamic facts of NOPL can be related to concepts of $\mathcal{M}$OISE organizations (roles, goals, missions, etc). But it is important to remark that the described use case situates NOPL instead of situating $\mathcal{M}$OISE. We are not situating goals, roles, missions, etc. Rather, we are defining how some states referred by the norms are constituted from the environment. NOPL norms are directed to states. When the goal of an obligation is `achieved(S,G,A)`, the norm expresses that the agents must see to them that the property *achieved(S,G,A)* holds. In the constitutive level, we simply define that, in some way, *achieved(S,G,A)* is constituted. But the fact of *achieved(S,G,A)* to mean that “*the goal G of the scheme S has been achieved by the agent A*” is neither part of the NOPL nor part of SAI. In the original *build-a-house* implementation, the agents must be able to make such relation between dynamic facts and $\mathcal{M}$OISE organisations. But, as discussed in Section 7.3.3.2, in the new implementation it is not required.

7.3.3.2 Analysing the institutionalised regulation

As we depart from an MAS that does not contain the notion of institution to the extended one where SAI is introduced, it is possible to compare both the systems, observing what is added by SAI in the new version. A first advantage observed when SAI is introduced is that the institution guides the agents towards to concretely act to comply with the system’s expectations. Consider, for example, the agent *CompanyA* receiving the obligation `obligation(companyA,N,play(CompanyA,bricklayer,‘‘hsh_group’’),D)`. In the original implementation, it is not clear how the agents must act in the environment to satisfy the goal of the obligation. In the new implementation, the constitutive rules make it explicit to the agents (cf. constitutive rule 3 in Figure 30).

A second advantage observed when SAI is introduced in the *build-a-house* application is that agents do not need to reason about the $\mathcal{M}$OISE organisation or even be aware of it, unless that makes sense in the particular application. For example, the obligation `obligation(companyA,N,play(CompanyA,bricklayer,“hsh_group”),D)` defines that *CompanyA* is obliged to produce the dynamic fact *play(CompanyA,bricklayer,“hsh_group”)* but it is not explicit that such fact corresponds to a role adoption. In the original implementation, the obliged agent must be aware of this relation between the obligation and its meaning in the organisation as the triggering of the required dynamic fact conditioned by the action of the agent informing to the organisational platform that he has adopted a role. In the new version of the MAS, company agents do not need to reason in terms of $\mathcal{M}$OISE concepts. In the aforementioned example, all that the agents must reason about is that the agent *CompanyA* must produce facts in the environment that count as the state-status function *play(CompanyA,bricklayer,“hsh_group”)* being constituted. The relation between the dynamic facts and the organisational state is made by the ORA4MAS machinery where the NOPL engine is enclosed.

A third advantage is a consequence of the first: agents cannot avoid the normative consequences of their actions (which in some application might be important, particularly in open system). In the original implementation, the hiring process consists of *Giacomo* asking the companies to adopt the corresponding roles when they win the auction. The company agents become then obliged to act to produce the dynamic fact $play(A,R,G)$. However, the companies can simply ignore the request and do not adopt the role (as they ought to in this application). Furthermore, if a company actually performs a required task (e.g to prepare the site) but does not informs it to the organisation (by producing the dynamic fact $achieved(S,G,A)$), the institution simply becomes inconsistent as it expects the agent to perform some task that is already done. In the new version, the dynamic facts that make the normative state to evolve are constituted from the environmental dynamics without requiring any action by the agents (the only required actions are those related to the application scenario). The evolution of the normative state therefore does not depend on the agents but on the environmental dynamics.

The fourth observed advantage is the simplification of the reasoning of the agents. Due to the possibility of modelling institutional consequences based on events and states, agents do not have to perform actions related to the institution (and naturally do not need to know how to do that). For example, the agent Giacomo performs 39 actions in the original example and this number was reduced to 19 in the new implementation (Table 5). The actions performed by the agent Giacomo in the new implementation are those related to the environment: instantiation of artifacts and handling of auctions. In this new implementation, however, Giacomo does not need to perform actions related to the institutional platform. For example, it needs neither to check the winners of the auctions nor to hire them (informing the roles that they should adopt). This reduction does not necessarily mean, however, either an improvement on system performance or a reduction in coding. It is essentially a conceptual change, as part of the code was moved from the agents program to the count-as rules. That moved code is better conceived as belonging to the institution than to the agents, so it is more coherent to program it outside of and independently from the agents. Our approach therefore appears to further improve the programming style available in a multi-agent oriented programming platform where the three distinct dimensions of a multi-agent system can be directly programmed.

Tabela 5 – Activities of agent *Giacomo*.

		Number of Executions	
		Original Example	New Implementation
1	Create the *GroupBoard* artifact	1	1
2	Adopt the *house owner* role	1	-
3	Commit to the *management_of_house_building* mission	1	-
4	Create the *SchemeBoard* artifact	1	1
5	Make the *GroupBoard* responsible for the *SchemeBoard*	1	-
6	Create the *AuctionArt* artifact	8	8
7	Check the deadlines of the auctions	1	1
8	Close the auctions	8	8
9	Check the auction winners	8	-
10	Hire the winners	8	-
11	Check if all the agents have adopted their roles	1	-
		39	19

7.3.3.3 Comparison with another approach

A different "count as" approach is applied to the *Build-a-house* example is also described in (BRITO; HÜBNER; BORDINI, 2013). The advantages observed in that case are the same as those previously in Section 7.3.3.2. But we can observe that, by introducing SAI, both the agents' reasoning and acting become even simpler. For example, the agent *CompanyA* performs 9 actions in the original implementation. In (BRITO; HÜBNER; BORDINI, 2013), this number is reduced to 5. Situating it with SAI, this number is reduced to 4 (Table 6). This additional reduction is possible because, in SAI, the constituted elements (the term *y* of count-as rules) are semantically linked to the regulative elements, that is not the case in (BRITO; HÜBNER; BORDINI, 2013). For this reason, in (BRITO; HÜBNER; BORDINI, 2013), the agents must to establish relations between the constituted elements and the norms while in SAI, this relation is explicit. Consider, for example, that *CompanyA* receives the obligation `obligation(companyA,N,achieved(Sch,plumbing_installed,companyA),D)` from the normative platform. By (BRITO; HÜBNER; BORDINI, 2013), we can specify the following count-as rule:

```
+installPlumbing[agent_name(companyA),artifact_name(housegui)]
 count-as
   achieved(Sch,plumbing_installed,companyA).
```

But the term *y* of the count-as rule and the goal (or the *aim*) of the obligation are not semantically related even they are syntactically equals. To know that it must produce the event `installPlumbing` in the environment to fulfil the obligation, *CompanyA* must know that the term *y* of the count-as rule will be taken by the organizational platform as the goal of the received obligation.

SAI makes explicit that the goals of the obligations are either event- or, as in the example, state-status functions. By receiving an obligation, the agents know that they must act in the environment to produce the required SFA. By the constitutive rules, the agents know how to act in this direction.

Tabela 6 – Actions of agent CompanyA

		Original example	Situated with Brito et al.[a]	Situated with SAI
1	Look for the group	✓	✓	✓
2	Look for auctions	✓	✓	✓
3	Submit bids to auctions	✓	✓	✓
4	Receive the contracting message	✓	-	-
5	Adopt a role	✓	-	-
6	Commitment to a mission corrsponding to the adopted role	✓	-	-
7	Act to fulfill obligations	✓	✓	✓
8	Inform the organisation about a goal achievement	✓	-	-
9	Relate goals to constituted elements	✓	✓	-
		9	5	4

---

[a] (BRITO; HÜBNER; BORDINI, 2013)

7.4 CONCLUSIONS

The *build-a-house* is an MAS originally conceived considering the regulation based on abstract concepts. But the notion of institutional reality is not considered in the original implementation. The relations between the abstract concepts used in the regulation and the environment are made by the agents themselves that must (i) know how to act in the environment to comply with their obligations, (ii) relate the normative goals with $\mathcal{M}$OISE concepts to properly manage the organisational infrastructure and (iii) act upon the organisational infrastructure. With SAI, the agents get rid of these requirements. The the facts that animate the normative dynamics are those composing the institutional reality in the system (i.e. the constitutive state). The constitutive rules make explicit how the environmental dynamics constitute the constitutive states considered by the normative program. The interpretation of the normative meaning of environmental facts is moved from the agents to the institutional programs (the constitutive specification, in this case).

8 CONCLUSIONS

The main contribution of this work is a model of institutional reality that arises from the environment to base the normative regulation in MAS. This general contribution results from the work developed to answer the questions posed in Section 1.2. The answers, described in the sequence, provide additional details about the contributions of this work.

1. *What are the proper abstractions to be used to represent (or the constructs to be used to specify) the elements of the institutional reality?*
   Answers to this question are given in Chapter 3. According to Searle's theory, the institutional reality exists because people assign functions to the elements of the brute reality that are not inherent to their natural virtues (i.e the *status functions*). Inspired by this conception, we consider *status functions assignments* as the building blocks of the institutional reality in artificial institutions. The institutional reality can be specified in terms of status functions. The constitution of status functions, that produces the status function assignments, is specified through constitutive rules.

2. *How are the different natures of the environmental elements taken into account within the institutional reality?*
   Answers to this question are given in Chapter 3. We defined that elements that have an institutional counterpart are the agents acting, the events occurring, and the states holding in the environment. The different natures of the environmental elements are captured in the institutional reality by the use of the three proposed kinds of status functions: agent-, event- , and state-status functions.

3. *Is there some difference between the institutional counterpart of agents, events, and other kinds of elements?*
   Answers to this question are given in chapters 3 and 4. It is important that the components of the institutional reality are distinguishable according to their environmental counterpart to consistently ground the institutional reality in the environment and to consistently couple the norms in the institutional reality. For example, it is important to clearly state what components of the institutional reality are the institutional counterparts of agents to (i) properly manage these elements within the institutional reality (e.g. keeping the counterpart only when the agent is actually participating in the system) and (ii) properly relate norms with the institutional reality (e.g. making clear to the normative layer what

are the elements that can be considered as the attribute (i.e. the bearer of an obligation).

4. *How are these differences represented and managed within the institutional reality?*
   Answers to this question are given in Chapter 3. First, the typing of the status functions makes distinguishable the different natures of the components of the institutional reality. Thus, the set of status functions of an institution is not just a vocabulary to be used to specify norms. Rather, the different kinds of status functions have clear, particular semantics as building blocks of the institutional reality.

   These differences are reflected, first, in the representations of status function assignments: agent-status function assignments are relations between an agent-status function and and environmental agent, event-status function assignments are ternary relations between an environmental event, an event-status function, and an environmental agent, and state-status function assignments are either relations between environmental states and state-status functions or freestanding assignments, where there is not an environmental counterpart for the status function. These differences are reflected also in the management of the institutional reality: agent-status function assignments hold only while the corresponding agent is participating in the system, state-status function assignment hold only while the corresponding state is holding in the environment, and event-status function assignments hold during a single step of the SAI history.

5. *How to base norms following different models on the same institutional reality?*
   The inspiration in Searle's theory lead us to consider that the path for basing norms, independent of how they are conceived, on the same institutional reality is to take the institutional reality as a *constitutive state*, that is the institutional counterpart of the environmental elements. A constitutive state provides the elements in terms of which norms can be specified, independent of how the dynamics of the regulation is conceived in the different normative models.

6. *Can the norms, as currently conceived in the literature, be based on an unified representation of institutional reality?*
   Answers to this question are given in Chapter 4. As seen in Section 2.2.1.1, norms are, in different ways, expressed by the ADICO elements. As shown in Section 4.1, the ADICO elements can be expressed in terms of status functions. Thus, an institutional reality expressed in

> terms of (constituted) status functions provides a unified vocabulary to specify norms that follow different models. But more than providing a vocabulary for the norms, the institutional reality as conceived by SAI endows such a vocabulary with a semantics – by typing the status functions – to be considered by the norms independent of the model they follow. Thus, for example, the ***aIm*** of any norm explicitly refers to the same kinds of element, namely event- and state-status functions.
>
> Basing norms on the institutional reality is not just a problem of specifying the norms. Rather, it is necessary to take into account that norms based on an interpretation of the environment are managed to regulate the elements that are under such interpretation. Although different normative models have a similar way to express the norms (i.e. the ADICO elements), they conceive different ways to manage norms, that cannot be generalised. But Section 4.2 generalises the strategy to base the different conceived normative managements on the same constitutive state. Applying such strategy requires to address particularly each normative model.

The previous questions are consequence of the main question motivating this work, that is: *how to represent the institutional reality, that arises from the environment to base the whole regulation in artificial institutions?* From all the developed work, we conclude that the institutional reality can be represented through status functions constituted from the environmental elements according to the specified by the constitutive rules. This conclusion is directly related to the inspiration taken from the work by John Searle. It is important to remark that our computational representation is one among other possible adaptations of that work to the MAS field.

These answers contribute to advance on the notion of *institution* in MAS. This notion is sometimes employed by works in the field. In this work, however, we contribute by defining, through the SAI model, a clearer notion of the components of an institution and their relation (that is not the only possible one): institutions enable the institutional reality (through the constitutive state in the case of SAI) and house the regulative representations (mainly norms in this work).

The proposed notion of institution adds power to design the social aspects of MAS as regulation and institutional reality are decoupled, each one modelling different, independent aspects of institutions. This is also an advantage for the development of tools to support the MAS programming as tools for deployment of both institutional reality and regulation can be developed independently but still connected to compose institutional infrastructures (cf. Chapter 5).

8.1 FUTURE WORK

By developing this thesis, some points have been observed that could be investigated in future work. Some of them are described below:

1. **Designing object-status functions.** The kinds of status functions considered in the current state of SAI are those essential to design norms. It is considered that agents acting, events occurring, and states holding in the environment are the elements that may have some function from the institutional perspective. But it seems to be also possible that *objects* existing in the environment have some function to be considered by the institution. For example, a signed paper sheet could count as a *contract*. Currently, it is possible to specify that some event counts as the event-status function of “sign a contract” or that some state counts as the state-status function of “signed contract”. But it is not possible to express, in the current SAI state, that a signed paper sheet counts as a “contract”. Having means to express these kind of constitution may be useful to specify the agents’ behaviour with respect to some objects in an abstract level, decoupled of, but constituted from, the environment.

2. **Handling of implicit constitutive representations in normative models.** Normative models usually consider norms to be specified in an abstract level but do not consider the notion of constitution. Such a notion is sometimes implicit in the normative models. For example, failure states of the NPL model seems to have some constitutive characteristics as it represents conditions that are seen from the normative perspective as undesirable states. SAI introduces an explicit constitutive level to support norms. Thus, it is possible to analyse normative models to check possible mixing of regulative and constitutive notions and to propose means to handle such mixing.

3. **Methodology to specify institutions.** The institutional design of application examples of chapters 6 and 7 was done in an *ad hoc* way. But given the different kinds elements involved in the design of institutions according to SAI and their relations, it seems interesting to look for methods to properly do such task. It is interesting, for example, to analyse, from a given use case, what is the best order for designing institutions, defining whether one should start by the norms, or by the status functions, etc. Additional aspects to be addressed are the extraction of status functions from the use cases, the choice for proper normative model considering the regulative requirements and the available set of status functions, etc.

4. **Institutionalising other social abstractions.** Besides norms, there are other social aspects in MAS, usually captured by metaphors inspired in human societies. There are, for example, organisations (DIGNUM; VÁZQUEZ-SALCEDA; DIGNUM, 2004; HÜBNER; SICHMAN; BOISSIER, 2007) and interaction protocols (SINGH, 2011; ZATELLI; HÜBNER, 2012). As well as norms, these abstractions define some expected behaviour from the agents. Also similar to norms, they are specified in an abstract level decoupled from the environment. For example, (SINGH, 2011) illustrates protocols being specified in terms of *debtor*, *creditor*, and *payment* but the protocol language does not specifies how these elements are constituted. As another example, the $\mathcal{M}$OISE model conceives organisations in terms of roles, goals, missions, groups, etc. but does not have means to specify how these elements are related to the environment. Given the "institutional" character of these social abstractions, it seems suitable to apply them to MAS in the context of institutions. Specially puzzling is the fact that some social abstractions proposed in current MAS research are not counterparts of individual agents, events, and states. For example, groups of $\mathcal{M}$OISE are social abstractions of sets of agents while missions are abstractions of sets of states to be produced following a given sequence. Thus, inserting these abstractions in institutions as conceived by SAI requires to define whether and how these abstractions, that refer to *sets* of environmental elements, can take part of an institutional reality composed as institutional counterparts of *individual* agents, events, and states from the environment.

8.2 RELATED PUBLICATIONS

Publications related to this thesis are enumerated in the sequence:

8.2.1 Conference papers

1. BRITO, M. de; HÜBNER, J. F. Institutional Situatedness in Multi-Agent Systems. In: **VIII Workshop-Escola de Sistemas de Agentes, seus Ambientes e Aplicações - WESAAC 2014**. 2014. p. 57–68.

   This paper analyses the state of the art on institutional reality and situatedness in MAS.

2. BRITO, M. de; HÜBNER, J. F.; BOISSIER, O. A conceptual model

for situated artificial institutions. In: BULLING, N. et al. (Ed.). **Computational Logic in Multi-Agent Systems**. Cham: Springer International Publishing, 2014, (Lecture Notes in Computer Science, v. 8624). p. 35–51.

This paper presents the conceptual model of SAI.

3. BRITO, M. de; HÜBNER, J. F.; BOISSIER O. Bringing constitutive dynamics to situated artificial institutions. In: PEREIRA, F. C. et al. (Ed.). **Progress in Artificial Intelligence - 17th Portuguese Conference on Artificial Intelligence, EPIA 2015, Coimbra, Portugal, September 8-11, 2015. Proceedings**. Cham: Springer International Publishing, 2015. (Lecture Notes in Computer Science, v. 9273), p. 624–637.

   This paper introduces the dynamics of the constitutive state.

4. BRITO, M. de; HÜBNER, J. F.; BOISSIER O. Coupling regulative and constitutive dimensions in situated artificial institutions. In: ROVATSOS, M.; VOUROS, G. A.; JULIÁN, V. (Ed.).**Multi-Agent Systems and Agreement Technologies - 13th European Conference, EUMAS 2015, and Third International Conference, AT 2015, Athens, Greece, December 17-18, 2015, Revised Selected Papers**. Cham: Springer International Publishing, 2015. (Lecture Notes in Computer Science, v. 9571), p. 318–334.

   This paper addressed the coupling of the normative regulation on the constitutive state.

5. BRITO, M. de et al. Situated artificial institution to support advanced regulation in the field of crisis management. In: DEMAZEAU, Y. et al. (Ed.). **Advances in Practical Applications of Agents, Multi-Agent Systems, and Sustainability: The PAAMS Collection - 13th International Conference, PAAMS 2015, Salamanca, Spain, June 3-4, 2015, Proceedings**. Cham: Springer International Publishing, 2015. (Lecture Notes in Computer Science, v. 9086), p. 66–79.

   This paper introduces the application of SAI in a collaboration plaftorm for crisis management.

6. BRITO, M. de et al. Situated regulation on a crisis management collaboration platform. In: DEMAZEAU, Y. et al. (Ed.). **Advances in Practical Applications of Agents, Multi-Agent Systems, and Sustainability: The PAAMS Collection - 13th International Conference, PAAMS 2015, Salamanca, Spain, June 3-4, 2015, Proceed-**

ings. Cham: Springer International Publishing, 2015. (Lecture Notes in Computer Science, v. 9086), p. 267–270.

This paper presents a simulator of a collaboration platform where the crisis managemnet is contextualized under SAI institutions.

7. BRITO, M. de et al. Institution artificielle située pour une aide à la régulation dans le cadre de la gestion de crises. In: VERCOUTER, L.; PICARD, G. (Ed.). 23es Journées Francophones sur les Systèmes Multi-Agents (JFSMA'15). Rennes, France: Cépaduès, 2015. p. 133–142.

   This paper addresses the application of SAI in a collaboration plaftorm for crisis management.

8.3 JOURNAL PAPERS

1. BRITO, M. de et al. Institution artificielle situÃ©e pour une aide à la régulation de la gestion de crises. **Revue d'Intelligence Artificielle**, v. 30, n. 1–2, p. 185–209, 2016.

2. BRITO, M. de et al. Supporting flexible regulation of crisis management by means of situated artificial institutions. **Frontiers of Information Technology & Electronic Engineering**, Zhejiang University Press & Springer, v. 17, n. 4, p. 309–324, 2016.

   Both the journal papers advance in the use of SAI in a crisis management scenario, presenting extended institutional specifications and extended analysis of the use of SAI in such scenario.

REFERÊNCIAS

ALDEWERELD, H. et al. Making norms concrete. In: HOEK, W. van der et al. (Ed.). **Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2010): Volume 1-3**. Richland, SC: IFAAMAS, 2010. p. 807–814.

ALECHINA, N. et al. Multi-cycle query caching in agent programming. In: DESJARDINS, M.; LITTMAN, M. (Ed.). **Proceedings of the Twenty-Seventh AAAI Conferernce on Artificial Intelligence (AAAI 2013)**. Bellevue, Washington: AAAI Press, 2013. v. 1, p. 32–38.

ANDRIGHETTO, G. et al. (Ed.). **Normative Multi-Agent Systems**, v. 4 de **Dagstuhl Follow-Ups**, (Dagstuhl Follow-Ups, v. 4). Dagstuhl, Germany: Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2013.

ARTIKIS, A.; PITT, J.; SERGOT, M. Animated specifications of computational societies. In: **Proceedings of the First international Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2002)**. New York, NY, USA: ACM Press, 2002. p. 1053–1062.

BALKE, T. et al. Norms in mas: Definitions and related concepts. In: ANDRIGHETTO, G. et al. (Ed.). **Normative Multi-Agent Systems**. Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2013, (Dagstuhl Follow-Ups, v. 4). p. 1–31.

BOELLA, G. et al. (Ed.). **Normative Multi-Agent Systems, 15.03. - 20.03.2009**, v. 09121 de **Dagstuhl Seminar Proceedings**, (Dagstuhl Seminar Proceedings, v. 09121). Dagstuhl, Germany: Schloss Dagstuhl - Leibniz-Zentrum für Informatik, Germany, 2009.

BOELLA, G.; TORRE, L. van der. An agent oriented ontology of social reality. In: **Procs. of Formal Ontologies in Information Systems(FOIS'2004)**. Amsterdam: IOS Press, 2004. p. 199–209.

BOELLA, G.; TORRE, L. van der. Constitutive norms in the design of normative multiagent systems. In: TONI, F.; TORRONI, P. (Ed.). **Computational Logic in Multi-Agent Systems**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, (Lecture Notes in Computer Science, v. 3900). p. 303–319. ISBN 978-3-540-33996-0.

BOELLA, G.; TORRE, L. van der. A logical architecture of a normative system. In: GOBLE, L.; MEYER, J.-J. (Ed.). **Deontic Logic and Artificial**

Normative Systems. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, (Lecture Notes in Computer Science, v. 4048). p. 24–35.

BOELLA, G.; TORRE, L. W. N. van der. Regulative and constitutive norms in normative multiagent systems. In: DUBOIS, D.; WELTY, C. A.; WILLIAMS, M.-A. (Ed.). **Principles of Knowledge Representation and Reasoning: Proceedings of the Ninth International Conference (KR2004), Whistler, Canada, June 2-5, 2004**. [S.l.]: AAAI Press, 2004. p. 255–266.

BOELLA, G.; TORRE, L. W. N. van der; VERHAGEN, H. (Ed.). **Normative Multi-agent Systems, 18.03. - 23.03.2007**, v. 07122 de **Dagstuhl Seminar Proceedings**, (Dagstuhl Seminar Proceedings, v. 07122). Dagstuhl, Germany: Internationales Begegnungs- und Forschungszentrum für Informatik (IBFI), Schloss Dagstuhl, Germany, 2007.

BOISSIER, O. et al. Multi-agent oriented programming with jacamo. **Science of Computer Programming**, Elsevier North-Holland, Inc., Amsterdam, The Netherlands, v. 78, n. 6, p. 747–761, 2013.

BOISSIER, O.; HÜBNER, J. F.; SICHMAN, J. S. Organization Oriented Programming: From Closed to Open Organizations. In: O'HARE, G. M. P. et al. (Ed.). **Engineering Societies in the Agents World VII**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, (Lecture Notes in Computer Science, v. 4457). p. 86–105.

BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. **The Unified Modeling Language User Guide**. [S.l.]: Addison-Wesley, 2005. (Object Technology Series).

BRACHMAN, R.; LEVESQUE, H. **Knowledge Representation and Reasoning**. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2004.

BRATMAN, M. E.; ISRAEL, D. J.; POLLACK, M. E. Plans and resource-bounded practical reasoning. **Computational Intelligence**, v. 4, p. 349–355, 1988.

BRITO, M. de; HÜBNER, J. F. Institutional Situatedness in Multi-Agent Systems. In: **VIII Workshop-Escola de Sistemas de Agentes, seus Ambientes e Aplicações - WESAAC 2014**. [S.l.: s.n.], 2014. p. 57–68.

BRITO, M. de; HÜBNER, J. F.; BOISSIER, O. A conceptual model for situated artificial institutions. In: BULLING, N. et al. (Ed.). **Computational**

Logic in Multi-Agent Systems. Cham: Springer International Publishing, 2014, (Lecture Notes in Computer Science, v. 8624). p. 35–51.

BRITO, M. de; HÜBNER, J. F.; BOISSIER, O. Bringing constitutive dynamics to situated artificial institutions. In: PEREIRA, F. C. et al. (Ed.). **Progress in Artificial Intelligence - 17th Portuguese Conference on Artificial Intelligence, EPIA 2015, Coimbra, Portugal, September 8-11, 2015. Proceedings**. Cham: Springer International Publishing, 2015. (Lecture Notes in Computer Science, v. 9273), p. 624–637.

BRITO, M. de; HÜBNER, J. F.; BOISSIER, O. Coupling regulative and constitutive dimensions in situated artificial institutions. In: ROVATSOS, M.; VOUROS, G. A.; JULIÁN, V. (Ed.). **Multi-Agent Systems and Agreement Technologies - 13th European Conference, EUMAS 2015, and Third International Conference, AT 2015, Athens, Greece, December 17-18, 2015, Revised Selected Papers**. Cham: Springer International Publishing, 2015. (Lecture Notes in Computer Science, v. 9571), p. 318–334.

BRITO, M. de; HÜBNER, J. F.; BORDINI, R. H. Programming institutional facts in multi-agent systems. In: ALDEWERELD, H.; SICHMAN, J. S. a. (Ed.). **Coordination, Organizations, Institutions, and Norms in Agent Systems VIII**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, (Lecture Notes in Computer Science, v. 7756). p. 158–173.

BRITO, M. de et al. Institution artificielle située pour une aide à la régulation dans le cadre de la gestion de crises. In: VERCOUTER, L.; PICARD, G. (Ed.). **23es Journées Francophones sur les Systèmes Multi-Agents (JFSMA'15)**. Rennes, France: Cépaduès, 2015. p. 133–142.

BRITO, M. de et al. Situated artificial institution to support advanced regulation in the field of crisis management. In: DEMAZEAU, Y. et al. (Ed.). **Advances in Practical Applications of Agents, Multi-Agent Systems, and Sustainability: The PAAMS Collection - 13th International Conference, PAAMS 2015, Salamanca, Spain, June 3-4, 2015, Proceedings**. Cham: Springer International Publishing, 2015. (Lecture Notes in Computer Science, v. 9086), p. 66–79.

BRITO, M. de et al. Situated regulation on a crisis management collaboration platform. In: DEMAZEAU, Y. et al. (Ed.). **Advances in Practical Applications of Agents, Multi-Agent Systems, and Sustainability: The PAAMS Collection - 13th International Conference, PAAMS 2015, Salamanca, Spain, June 3-4, 2015, Proceedings**. Cham: Springer International Publishing, 2015. (Lecture Notes in Computer Science, v. 9086), p. 267–270.

BRITO, M. de et al. Institution artificielle située pour une aide à la régulation de la gestion de crises. **Revue d'Intelligence Artificielle**, v. 30, n. 1-2, p. 185–209, 2016.

BRITO, M. de et al. Supporting flexible regulation of crisis management by means of situated artificial institutions. **Frontiers of Information Technology & Electronic Engineering**, Zhejiang University Press & Springer, v. 17, n. 4, p. 309–324, 2016.

BROERSEN, J.; TORRE, L. van der. Ten problems of deontic logic and normative reasoning in computer science. In: BEZHANISHVILI, N.; GORANKO, V. (Ed.). **Lectures on Logic and Computation**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, (Lecture Notes in Computer Science, v. 7388). p. 55–88.

BROERSEN, J. M. et al. Normative reasoning and consequence. In: ANDRIGHETTO, G. et al. (Ed.). **Normative Multi-Agent Systems**. Dagstuhl, Germany: Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2013, (Dagstuhl Follow-Ups, v. 4). p. 33–70.

CAMPOS, J. et al. Formalising Situatedness and Adaptation in Electronic Institutions. In: HÜBNER, J. et al. (Ed.). **Coordination, Organizations, Institutions and Norms in Agent Systems IV**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, (Lecture Notes in Computer Science, v. 5428). p. 126–139.

CARDOSO, H. L.; OLIVEIRA, E. C. Institutional reality and norms: Specifying and monitoring agent organizations. **Int. J. Cooperative Inf. Syst.**, v. 16, n. 1, p. 67–95, 2007.

CARDOSO, H. L.; OLIVEIRA, E. C. A context-based institutional normative environment. In: HÜBNER, J. F. et al. (Ed.). **Coordination, Organizations, Institutions and Norms in Agent Systems IV, COIN 2008 International Workshops, COIN@AAMAS 2008, Estoril, Portugal, May 12, 2008. COIN@AAAI 2008, Chicago, USA, July 14, 2008. Revised Selected Papers**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008. (Lecture Notes in Computer Science, v. 5428), p. 140–155.

CASSANDRAS, C. G.; LAFORTUNE, S. **Introduction to Discrete Event Systems**. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006.

CASTELFRANCHI, C. Engineering social order. In: OMICINI, A.; TOLKSDORF, R.; ZAMBONELLI, F. (Ed.). **Engineering Societies in the Agent World, First International Workshop, ESAW 2000, Berlin,**

Germany, August 21, 2000, Revised Papers. Berlin, Heidelberg: Springer Berlin Heidelberg, 2000. (Lecture Notes in Computer Science, v. 1972), p. 1–18.

CLIFFE, O.; VOS, M. D.; PADGET, J. Answer set programming for representing and reasoning about virtual institutions. In: INOUE, K.; SATOH, K.; TONI, F. (Ed.). **Computational Logic in Multi-Agent Systems**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, (Lecture Notes in Computer Science, v. 4371). p. 60–79.

COMMONS, J. R. **Institutional Economics – Its Place in Political Economy**. [S.l.]: The Macmillan Company, 1934. (Institutional Economics Series, v. 1).

COSTA, A. C. da R.; DIMURO, G. P. Semantical concepts for a formal structural dynamics of situated multiagent systems. In: **Coordination, Organizations, Institutions, and Norms in Agent Systems III, COIN 2007, International Workshops COIN@AAMAS 2007, Honolulu, HI, USA, May 14, 2007, COIN@MALLOW 2007, Durham, UK, September 3-4, 2007, Revised Selected Papers**. [S.l.: s.n.], 2007. p. 139–154.

COSTA, A. C. da R.; DIMURO, G. P. A minimal dynamical organization model, and how it supports intensional organization processes. In: DIGNUM, V. (Ed.). **Handbook of Research on Multi-Agent Systems: Semantics and Dynamics of Organizational Models**. 1. ed. Hershey, USA: Information Science Reference, 2009. cap. 17, p. 419–445.

CRAWFORD, S. E. S.; OSTROM, E. A Grammar of Institutions. **The American Political Science Review**, v. 89, n. 3, p. 582–600, 1995.

CRIADO, N.; ARGENTE, E.; BOTTI, V. J. Open issues for normative multi-agent systems. **AI Commun.**, v. 24, n. 3, p. 233–264, 2011.

DASTANI, M.; DIGNUM, V.; DIGNUM, F. Role-assignment in open agent societies. In: **The Second International Joint Conference on Autonomous Agents & Multiagent Systems, AAMAS 2003, July 14-18, 2003, Melbourne, Victoria, Australia, Proceedings**. New York, NY, USA: ACM, 2003. p. 489–496.

DASTANI, M. et al. Knowledge Representation for Agents and Multi-Agent Systems. In: MEYER, J.-J. C.; BROERSEN, J. (Ed.). Berlin, Heidelberg: Springer-Verlag, 2009. cap. Normative Multi-agent Programs and Their Logics, p. 16–31.

DASTANI, M.; TORRE, L. W. N. van der; YORKE-SMITH, N. Monitoring interaction in organisations. In: **Coordination, Organizations, Institutions, and Norms in Agent Systems VIII - 14th International Workshop, COIN 2012, Held Co-located with AAMAS 2012, Valencia, Spain, June 5, 2012, Revised Selected Papers**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012. p. 17–34.

DECKER, K. Taems: A framework for environment centered analysis & design of coordination mechanisms. In: O'HARE, G.; JENNINGS, N. (Ed.). **Foundations of Distributed Artificial Intelligence**. New York, USA: Wiley Inter-Science, 1996. cap. 16, p. 429–448.

DEMAZEAU, Y.; COSTA, A. C. R. Populations and Organizations in Open Multi-agent Systems. In: **Proceedings of the National Symposium on Parallel amd Distributed Artificial Intelligence (PDAI'96)**. Hyderabad, India: [s.n.], 1996.

DEMAZEAU, Y. et al. (Ed.). **Advances in Practical Applications of Agents, Multi-Agent Systems, and Sustainability: The PAAMS Collection - 13th International Conference, PAAMS 2015, Salamanca, Spain, June 3-4, 2015, Proceedings**, v. 9086 de **Lecture Notes in Computer Science**, (Lecture Notes in Computer Science, v. 9086). Cham: Springer International Publishing, 2015.

DIGNUM, V. **A model for organizational interaction: based on agents, founded in logic**. Tese (Doutorado), 2004.

DIGNUM, V.; VÁZQUEZ-SALCEDA, J.; DIGNUM, F. A model of almost everything: Norms, structure and ontologies in agent organizations. In: **3rd International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2004), 19-23 August 2004, New York, NY, USA**. Washington, DC, USA: IEEE Computer Society, 2004. p. 1498–1499.

ESTEVA, M. et al. On the Formal Specifications of Electronic Institutions. In: DIGNUM, F.; SIERRA, C. (Ed.). **AgentLink**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001. (Lecture Notes in Computer Science, v. 1991), p. 126–147. ISBN 3-540-41671-4.

FARJAM, M.; KIRCHKAMP, O. **Bubbles in Hybrid Markets - How Expectations about Algorithmic Trading Affect Human Trading**. [S.l.], 2015.

FASLI, M. Accounting for social order in multi-agent systems: Preliminary report. In: **2004 IEEE/WIC/ACM International Conference on**

Intelligent Agent Technology (IAT 2004), 20-24 September 2004, Beijing, China. [S.l.]: IEEE Computer Society, 2004. p. 204–210.

FERBER, J.; GUTKNECHT, O.; MICHEL, F. From Agents to Organizations: An Organizational View of Multi-agent Systems. In: **Agent-Oriented Software Engineering IV: 4th InternationalWorkshop, AOSE 2003, Melbourne, Australia, July 15, 2003. Revised Papers**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004. p. 214–230.

FERBER, J.; MICHEL, F.; BÁEZ-BARRANCO, J.-A. AGRE: Integrating Environments with Organizations. In: **Environments for Multi-Agent Systems: First International Workshop, E4MAS 2004, New York, NY, July 19, 2004, Revised Selected Papers**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004. p. 48–56.

FIGUEIREDO, K. da S.; SILVA, V. T. da. An algorithm to identify conflicts between norms and values. In: **Coordination, Organizations, Institutions, and Norms in Agent Systems IX - COIN 2013 International Workshops, COIN@AAMAS, St. Paul, MN, USA, May 6, 2013, COIN@PRIMA, Dunedin, New Zealand, December 3, 2013, Revised Selected Papers**. Cham: Springer International Publishing, 2013. p. 259–274.

FIGUEIREDO, K. da S.; SILVA, V. Torres da; BRAGA, C. de O. Modeling norms in multi-agent systems with normml. In: VOS, M. D. et al. (Ed.). **Coordination, Organizations, Institutions, and Norms in Agent Systems VI**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, (Lecture Notes in Computer Science, v. 6541). p. 39–57.

FIROZABADI, B. S.; TORRE, L. W. N. van der. Formal models of control systems. In: **ECAI**. [S.l.: s.n.], 1998. p. 317–318.

FORNARA, N.; COLOMBETTI, M. Specifying and Enforcing Norms in Artificial Institutions. In: DUNIN-KEPLICZ, B.; OMICINI, A.; PADGET, J. A. (Ed.). **EUMAS**. [S.l.]: CEUR-WS.org, 2006. (CEUR Workshop Proceedings, v. 223).

FORNARA, N.; COLOMBETTI, M. Specifying artificial institutions in the event calculus. In: ____. **Handbook of Research on Multi-Agent Systems: Semantics and Dynamics of Organizational Models**. 1. ed. Hershey, USA: Information Science Reference, 2009. cap. 14, p. 335–366.

FORNARA, N.; VIGANÒ, F.; COLOMBETTI, M. Agent Communication and Institutional Reality. In: **Agent Communication, International Workshop on Agent Communication, AC 2004, New York, NY, USA,**

July 19, 2004, Revised Selected and Invited Papers. [S.l.: s.n.], 2004. p. 1–17.

FORNARA, N.; VIGANÒ, F.; COLOMBETTI, M. Agent communication and artificial institutions. **Autonomous Agents and Multi-Agent Systems**, v. 14, n. 2, 2007.

FORNARA, N. et al. Artificial institutions: a model of institutional reality for open multiagent systems. **Artificial Intelligence and Law**, v. 16, n. 1, 2008.

FRANKE, J.; CHAROY, F. Design of a collaborative disaster response process management system. In: **9th International conference on the design of Cooperative Systems**. [S.l.: s.n.], 2010.

FRANKLIN, S.; GRAESSER, A. Is it an agent, or just a program?: A taxonomy for autonomous agents. In: MÜLLER, J. P.; WOOLDRIDGE, M. J.; JENNINGS, N. R. (Ed.). **Intelligent Agents III Agent Theories, Architectures, and Languages: ECAI'96 Workshop (ATAL) Budapest, Hungary, August 12–13, 1996 Proceedings**. Berlin, Heidelberg: Springer Berlin Heidelberg, 1997, (Lecture Notes in Computer Science, v. 1193). p. 21–35.

GATEAU, B. **Modélisation et Supervision d'Institutions Multi-Agents**. Tese (Doutorado) — École Nationale Supéurieure des Mines de Saint Étienne, 2007.

GEIGER, R. S. The social roles of bots and assisted editing programs. In: RIEHLE, D.; BRUCKMAN, A. (Ed.). **Proceedings of the 2009 International Symposium on Wikis, 2009, Orlando, Florida, USA, October 25-27, 2009**. New York, NY, USA: ACM, 2009.

GEIGER, R. S. The lives of bots. In: LOVINK, G.; TKACZ, N. (Ed.). **Critical Point of View: A Wikipedia Reader**. Amsterdam: Institute of Network Cultures, 2011. p. 79–93.

GROSSI, D.; DIGNUM, F. From abstract to concrete norms in agent institutions. In: HINCHEY, M. et al. (Ed.). **Formal Approaches to Agent-Based Systems**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, (Lecture Notes in Computer Science, v. 3228). p. 12–29.

GROSSI, D.; GABBAY, D.; TORRE, L. Specification and verification of multi-agent systems. In: ____. Boston, MA: Springer US, 2010. cap. The Norm Implementation Problem in Normative Multi-Agent Systems.

HART, H. et al. **The Concept of Law**. Oxford: OUP Oxford, 2012. (Clarendon Law Series).

HELLEBOOGH, A. et al. Modeling dynamic environments in multi-agent simulation. **Autonomous Agents and Multi-Agent Systems**, v. 14, n. 1, p. 87–116, 2006.

HINDRIKS, F. But where is the university? **Dialectica**, Blackwell Publishing Ltd, v. 66, n. 1, p. 93–113, 2012.

HODGSON, G. M. What are institutions. **Journal of Economic Issues**, n. 1, p. 1–25, 2006.

HOLLANDER, C. D.; WU, A. S. The current state of normative agent-based systems. **J. Artificial Societies and Social Simulation**, v. 14, n. 2, 2011.

HÜBNER, J. F.; BOISSIER, O.; BORDINI, R. H. A normative organisation programming language for organisation management infrastructures. In: PADGET, J. et al. (Ed.). **Coordination, Organizations, Institutions and Norms in Agent Systems V**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, (Lecture Notes in Computer Science, v. 6069). p. 114–129.

HÜBNER, J. F.; BOISSIER, O.; BORDINI, R. H. A normative programming language for multi-agent organisations. **Annals of Mathematics and Artificial Intelligence**, v. 62, n. 1-2, 2011.

HÜBNER, J. F. et al. Instrumenting multi-agent organisations with organisational artifacts and agents. **Autonomous Agents and Multi-Agent Systems**, v. 20, n. 3, p. 369–400, abr. 2009. ISSN 1387-2532.

HÜBNER, J. F.; SICHMAN, J. S.; BOISSIER, O. A model for the structural, functional, and deontic specification of organizations in multiagent systems. In: BITTENCOURT, G.; RAMALHO, G. L. (Ed.). **Proceedings of the 16th Brazilian Symposium on Artificial Intelligence (SBIA'02)**. Berlin: Springer, 2002. (LNAI, v. 2507), p. 118–128.

HÜBNER, J. F.; SICHMAN, J. S.; BOISSIER, O. Developing organised multiagent systems using the MOISE+ model: Programming issues at the system and agent levels. **IJAOSE**, v. 1, n. 3/4, p. 370–395, 2007.

JONES, A.; SERGOT, M. A formal characterisation of institutionalised power. **Logic Journal of IGPL**, Oxford Univ Press, v. 4, n. 3, p. 427–443, 1996.

JONES, A. J. I.; SERGOT, M. Deontic logic in computer science. In: MEYER, J.-J. C.; WIERINGA, R. J. (Ed.). Chichester, UK: John Wiley and Sons Ltd., 1993. cap. On the Characterization of Law and Computer Systems: The Normative Systems Perspective, p. 275–307.

KOLSKI, C. et al. Interactive surfaces, tangible interaction: Perspectives for risk management. In: ____. **Risk Management in Life-Critical Systems**. [S.l.]: John Wiley & Sons, Inc., 2015. p. 351–373.

KUBICKI, S.; LEPREUX, S.; KOLSKI, C. Rfid-driven situation awareness on tangisense, a table interacting with tangible objects. **Personal and Ubiquitous Computing**, Springer-Verlag, v. 16, n. 8, p. 1079–1094, 2012.

LESSER, V. R. et al. Evolution of the gpgp/tæms domain-independent coordination framework. **Autonomous Agents and Multi-Agent Systems**, v. 9, n. 1-2, p. 87–143, 2004.

LOPEZ, F. L. y; LUCK, M. Towards a model of the dynamics of normative multi-agent systems. In: **Proc. of the Int. Workshop on Regulated Agent-Based Social Systems: Theories and Applications**. [S.l.]: Universität, 2002. (Mitteilung, v. 318), p. 175–194.

LÓPEZ, F. L. y; LUCK, M.; D'INVERNO, M. Constraining autonomy through norms. In: **The First International Joint Conference on Autonomous Agents & Multiagent Systems, AAMAS 2002, July 15-19, 2002, Bologna, Italy, Proceedings**. New York, NY, USA: ACM, 2002. p. 674–68,1.

LÓPEZ, F. L. y; LUCK, M.; D'INVERNO, M. A normative framework for agent-based systems. In: BOELLA, G.; TORRE, L. W. N. van der; VERHAGEN, H. (Ed.). **Normative Multi-agent Systems**. Dagstuhl, Germany: Internationales Begegnungs- und Forschungszentrum für Informatik (IBFI), Schloss Dagstuhl, Germany, 2007. (Dagstuhl Seminar Proceedings, v. 07122).

LÓPEZ, F. López y; LUCK, M. Modelling norms for autonomous agents. In: **4th Mexican International Conference on Computer Science (ENC 2003), 8-12 September 2003, Apizaco, Mexico**. Los Alamitos, CA: IEEE Computer Society Press, 2003. p. 238–245.

LÓPEZ, F. López y; LUCK, M.; D'INVERNO, M. Normative agent reasoning in dynamic societies. In: **3rd International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2004), 19-23 August 2004, New York, NY, USA**. Washington, DC, USA: IEEE Computer Society, 2004. p. 732–739.

LUCK, M. et al. Agent technology: computing as interaction (a roadmap for agent based computing). University of Southampton, 2005.

MAES, P. Artificial life meets entertainment: lifelike autonomous agents. **Communications of the ACM**, v. 38, n. 11, p. 108–114, 1995.

MALUCELLI, A.; CARDOSO, H. L.; OLIVEIRA, E. C. Enriching a MAS environment with institutional services. In: **Environments for Multi-Agent Systems II, Second International Workshop,E4MAS 2005, Utrecht, The Netherlands, July 25, 2005, Selected Revised and Invited Papers**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005. p. 105–120.

MARRA, A.; KLEIN, D. Logic and ethics: An integrated model for norms, intentions and actions. In: **Logic, Rationality, and Interaction - 5th International Workshop, LORI 2015 Taipei, Taiwan, October 28-31, 2015, Proceedings**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2015. p. 268–281.

MILLER, S. Social institutions. In: ZALTA, E. N. (Ed.). **The Stanford Encyclopedia of Philosophy**. Fall 2012. [S.l.: s.n.], 2012.

MOSES, Y.; TENNENHOLTZ, M. Artificial social systems. **Computers and Artificial Intelligence**, v. 14, n. 6, 1995.

NORTH, D. C. Economic performance through time. **American Economic Review**, v. 84, p. 359–368, 1994.

OREN, N. et al. Towards a formalisation of electronic contracting environments. In: HÜBNER, J. F. et al. (Ed.). **Coordination, Organizations, Institutions and Norms in Agent Systems IV**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, (Lecture Notes in Computer Science, v. 5428). p. 156–171.

PACHECO, N. C. **Using Norms to Control Open Multi-Agent Systems**. Tese (Doutorado) — Universidad Politécnica de Valencia, Valencia, Spain, 2012.

PANAGIOTIDI, S.; ÁLVAREZ-NAPAGAO, S.; VÁZQUEZ-SALCEDA, J. Towards the norm-aware agent: Bridging the gap between deontic specifications and practical mechanisms for norm monitoring and norm-aware planning. In: **Coordination, Organizations, Institutions, and Norms in Agent Systems IX - COIN 2013 International Workshops, COIN@AAMAS, St. Paul, MN, USA, May 6, 2013, COIN@PRIMA, Dunedin, New Zealand, December 3, 2013, Revised Selected Papers**. Cham: Springer International Publishing, 2013. p. 346–363.

PIUNTI, M. **Designing and Programming Organizational Infrastructures for Agents Situated in Artifact-based Environments**. Tese (Doutorado) — Universit‘ di Bologna, Bologna, Italy, 2009.

PIUNTI, M. et al. Embodied organizations: a unifying perspective in programming agents, organizations and environments. In: **Proceedings of The Multi-Agent Logics, Languages, and Organisations Federated Workshops (MALLOW 2010)**. [S.l.]: CEUR-WS.org, 2010. (CEUR Workshop Proceedings, v. 627).

RICCI, A.; PIUNTI, M.; VIROLI, M. Environment programming in multi-agent systems: an artifact-based perspective. **Autonomous Agents and Multi-Agent Systems**, Springer US, v. 23, n. 2, p. 158–192, 2011.

RUITER, D. W. A basic classification of legal institutions. **Ratio Juris**, Blackwell, v. 10, n. 4, p. 357–371, December 1997.

RUSSELL, S.; NORVIG, P. **Artificial Intelligence: A Modern Approach**. 2nd edition. ed. Englewood Cliffs, NJ, USA: Prentice-Hall, 2003.

SEARLE, J. **The Construction of Social Reality**. [S.l.]: Free Press, 1995.

SEARLE, J. **Making the Social World:The Structure of Human Civilization**. [S.l.]: Oxford University Press, 2009.

SINGH, M. P. An ontology for commitments in multiagent systems. **Artif. Intell. Law**, v. 7, n. 1, p. 97–113, 1999.

SINGH, M. P. Information-driven interaction-oriented programming: Bspl, the blindingly simple protocol language. In: **10th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2011), Taipei, Taiwan, May 2-6, 2011, Volume 1-3**. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems, 2011. p. 491–498.

SMITH, B. John searle: From speech acts to social reality. In: SMITH, B. (Ed.). **John Searle**. [S.l.]: Cambridge University Press, 2003. p. 1–33.

SMITH, B.; SEARLE, J. The Construction of Social Reality: An Exchange. **American Journal of Economics and Sociology**, v. 62, n. 1, p. 285–309, 2003.

TESSOP, R. K. **Gestion de l’ouverture au sein d’organisations multi-agents: une approche basée sur des artefacts organisationnels**. Tese (Doutorado) — École Nationale Supéurieure des Mines de Saint-Étienne, 2011.

TESTERINK, B. **Norms for Distributed Organizations – Syntax, Semantics and Interpreter** . Dissertação (Mestrado) — Faculty of Science – Utrecht University, Utrecht, The Netherlands, 2012.

THÉVIN, L. et al. Un système multi-agent normatif hybride pour l'interaction mixte. **Revue d'Intelligence Artificielle**, v. 29, n. 3-4, p. 453–482, 2015.

TINNEMEIER, N. A. M. et al. Programming Normative Artifacts with Declarative Obligations and Prohibitions. **Web Intelligence and Intelligent Agent Technology, IEEE/WIC/ACM International Conference on**, IEEE Computer Society, Los Alamitos, CA, USA, v. 2, p. 145–152, 2009.

TORRE, L. W. N. van der et al. Specifying multiagent organizations. In: **Deontic Logic in Computer Science, 7th International Workshop on Deontic Logic in Computer Science, DEON 2004, Madeira, Portugal, May 26-28, 2004. Proceedings**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004. p. 243–257.

VASCONCELOS, W. W.; KOLLINGBAUM, M. J.; NORMAN, T. J. Normative conflict resolution in multi-agent systems. **Autonomous Agents and Multi-Agent Systems**, Springer US, v. 19, n. 2, p. 124–152, 2009.

VÁZQUEZ-SALCEDA, J.; ALDEWERELD, H.; DIGNUM, F. Implementing norms in multiagent systems. In: **Multiagent System Technologies, Second German Conference, MATES 2004, Erfurt, Germany, September 29-30, 2004, Proceedings**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004. p. 313–327.

VÁZQUEZ-SALCEDA, J.; ALDEWERELD, H.; DIGNUM, F. Norms in multiagent systems: some implementation guidelines. In: **In Proceedings of the Second European Workshop on Multi-Agent Systems (EUMAS**. [S.l.: s.n.], 2004. p. 737–748.

VIGANÒ, F.; COLOMBETTI, M. Specification and verification of institutions through status functions. In: NORIEGA, P. et al. (Ed.). **Coordination, Organizations, Institutions, and Norms in Agent Systems II**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, (Lecture Notes in Computer Science, v. 4386). p. 115–129.

VIGANÒ, F.; COLOMBETTI, M. Model checking norms and sanctions in institutions. In: SICHMAN, J. S. a. et al. (Ed.). **Coordination, Organizations, Institutions, and Norms in Agent Systems III**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, (Lecture Notes in Computer Science, v. 4870). p. 316–329.

VOS, M. D.; BALKE, T.; SATOH, K. Combining event-and state-based norms. In: **International conference on Autonomous Agents and Multi-Agent Systems, AAMAS '13, Saint Paul, MN, USA, May 6-10, 2013**. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems, 2013. p. 1157–1158.

WEYNS, D.; OMICINI, A.; ODELL, J. Environment as a first-class abstraction in multiagent systems. **Autonomous Agents and Multi-Agent Systems**, Citeseer, v. 14, n. 1, p. 5–30, 2007.

WOOLDRIDGE, M.; JENNINGS, N. Agent theories, architectures, and languages: A survey. In: WOOLDRIDGE, M.; JENNINGS, N. (Ed.). **Intelligent Agents: ECAI-94 Workshop on Agent Theories, Architectures, and Languages Amsterdam, The Netherlands August 8–9, 1994 Proceedings**. Berlin, Heidelberg: Springer Berlin Heidelberg, 1995, (Lecture Notes in Computer Science, v. 890). p. 1–39.

ZATELLI, M. R.; HÜBNER, J. F. A unified interaction model with agent, organisation, and environment. In: **Anais do IX Encontro Nacional de Inteligncia Artificial (ENIA@BRACIS)**. Curitiba, Brazil: [s.n.], 2012.

APÊNDICE A – NOPL Specification of the *Build-a-house* example

The NOPL program that runs in the group board artifact of the *Build-a-house* example is shown below:

```
scope organisation(house_group)

  role_cardinality(house_owner,1,1).
  role_cardinality(painter,1,1).
  role_cardinality(bricklayer,1,2).
  role_cardinality(window_fitter,1,1).
  role_cardinality(plumber,1,1).
  role_cardinality(electrician,1,1).
  role_cardinality(roofer,1,1).
  role_cardinality(site_prep_contractor,1,1).
  role_cardinality(door_fitter,1,1).

  compatible(building_company,building_company,gr_inst).

  rplayers(R,G,V)    :- .count(play(_,R,G),V).
  well_formed(G) :-
     rplayers(house_owner,G,Vhouse_owner) & Vhouse_owner >= 1 & Vhouse_owner <= 1 &
     rplayers(painter,G,Vpainter) & Vpainter >= 1 & Vpainter <= 1 &
     rplayers(bricklayer,G,Vbricklayer) & Vbricklayer >= 1 & Vbricklayer <= 2 &
     rplayers(window_fitter,G,Vwindow_fitter) & Vwindow_fitter >= 1 & Vwindow_fitter <= 1 &
     rplayers(plumber,G,Vplumber) & Vplumber >= 1 & Vplumber <= 1 &
     rplayers(electrician,G,Velectrician) & Velectrician >= 1 & Velectrician <= 1 &
     rplayers(roofer,G,Vroofer) & Vroofer >= 1 & Vroofer <= 1 &
     rplayers(site_prep_contractor,G,Vsite_prep_contractor) & Vsite_prep_contractor >= 1 & Vsite_prep_contrac
     rplayers(door_fitter,G,Vdoor_fitter) & Vdoor_fitter >= 1 & Vdoor_fitter <= 1 &
     .findall(GInst, subgroup(GInst,_,G), ListSubgroups) & all_subgroups_well_formed(ListSubgroups).
  all_subgroups_well_formed([]).
  all_subgroups_well_formed([H|T]) :- subgroup_well_formed(H) & all_subgroups_well_formed(T).


  norm role_in_group:
          play(Agt,R,Gr) &
          group_id(Gr) &
          not role_cardinality(R,_,_)
       -> fail(role_in_group(Agt,R,Gr)).
  norm role_cardinality:
          group_id(Gr) &
          role_cardinality(R,_,RMax) &
          rplayers(R,Gr,RP) &
          RP > RMax
       -> fail(role_cardinality(R,Gr,RP,RMax)).
  norm role_compatibility:
          play(Agt,R1,Gr) &
          play(Agt,R2,Gr) &
          group_id(Gr) &
          R1 < R2 &
          not fcompatible(R1,R2,gr_inst)
       -> fail(role_compatibility(R1,R2,Gr)).
  norm well_formed_responsible:
          responsible(Gr,S) &
          not monitor_scheme(S) &
          not well_formed(Gr)
       -> fail(well_formed_responsible(Gr)).
  norm subgroup_in_group:
          group_id(Gr) &
          subgroup(G,GT,Gr) &
          not subgroup_cardinality(GT,_,_)
       -> fail(subgroup_in_group(G,GT,Gr)).
  norm subgroup_cardinality:
          group_id(Gr) &
          subgroup_cardinality(SG,_,SGMax) &
          .count(subgroup(_,SG,Gr),SGP) &
          SGP > SGMax
       -> fail(subgroup_cardinality(SG,Gr,SGP,SGMax)).
```

The NOPL program that runs in the scheme board artifact of the *Build-a-house* example is shown below:

```
scope scheme(build_house_sch)

 // ** Facts from OS

 // mission_cardinality(mission id, min, max)
 mission_cardinality(build_walls,1,1).
 mission_cardinality(fit_doors,1,1).
 mission_cardinality(fit_windows,1,1).
 mission_cardinality(management_of_house_building,1,1).
 mission_cardinality(build_roof,1,1).
 mission_cardinality(paint_house,1,1).
 mission_cardinality(install_electrical_system,1,1).
 mission_cardinality(prepare_site,1,1).
 mission_cardinality(lay_floors,1,1).
 mission_cardinality(install_plumbing,1,1).

 // mission_role(mission id, role id)
 mission_role(management_of_house_building,house_owner).
 mission_role(prepare_site,site_prep_contractor).
 mission_role(lay_floors,bricklayer).
 mission_role(build_walls,bricklayer).
 mission_role(build_roof,roofer).
 mission_role(fit_windows,window_fitter).
 mission_role(fit_doors,door_fitter).
 mission_role(install_plumbing,plumber).
 mission_role(install_electrical_system,electrician).
 mission_role(paint_house,painter).

 // mission_goal(mission id, goal id)
 mission_goal(build_walls,walls_built).
 mission_goal(fit_doors,doors_fitted).
 mission_goal(fit_windows,windows_fitted).
 mission_goal(management_of_house_building,house_built).
 mission_goal(build_roof,roof_built).
 mission_goal(paint_house,exterior_painted).
 mission_goal(paint_house,interior_painted).
 mission_goal(install_electrical_system,electrical_system_installed).
 mission_goal(prepare_site,site_prepared).
 mission_goal(lay_floors,floors_laid).
 mission_goal(install_plumbing,plumbing_installed).

 // goal(missions, goal id, dependence (on goal statisfaction), type, #ags to satisfy, ttf)
 goal([paint_house],exterior_painted,dep(and,[rwd]),achievement,all,‘20 minutes‘).
 goal([prepare_site],site_prepared,dep(and,[]),achievement,all,‘20 minutes‘).
 goal([install_plumbing],plumbing_installed,dep(and,[rwd]),achievement,all,‘20 minutes‘).
 goal([management_of_house_building],house_built,dep(and,[interior_painted]),
                                                      achievement,all,‘1 year‘).
 goal([build_roof],roof_built,dep(and,[walls_built]),achievement,all,‘30 minutes‘).
 goal([build_walls],walls_built,dep(and,[floors_laid]),achievement,all,‘40 minutes‘).
 goal([],pee,dep(and,[plumbing_installed, electrical_system_installed,
                                     exterior_painted]),achievement,0,‘1 year‘).
 goal([lay_floors],floors_laid,dep(and,[site_prepared]),achievement,all,‘25 minutes‘).
 goal([fit_windows],windows_fitted,dep(and,[walls_built]),achievement,all,‘10 minutes‘).
 goal([paint_house],interior_painted,dep(and,[pee]),achievement,all,‘30 minutes‘).
 goal([],rwd,dep(and,[roof_built, windows_fitted, doors_fitted]),achievement,0,‘1 year‘).
 goal([install_electrical_system],electrical_system_installed,
                                         dep(and,[rwd]),achievement,all,‘20 minutes‘).
 goal([fit_doors],doors_fitted,dep(and,[walls_built]),achievement,all,‘10 minutes‘).
```

```
super_goal(pee, exterior_painted).
super_goal(house_built, site_prepared).
super_goal(pee, plumbing_installed).
super_goal(rwd, roof_built).
super_goal(house_built, walls_built).
super_goal(house_built, pee).
super_goal(house_built, floors_laid).
super_goal(rwd, windows_fitted).
super_goal(house_built, interior_painted).
super_goal(house_built, rwd).
super_goal(pee, electrical_system_installed).
super_goal(rwd, doors_fitted).

// ** Rules
mplayers(M,S,V) :- .count(committed(_,M,S),V).
well_formed(S) :-
   (mission_accomplished(S,build_walls) | mplayers(build_walls,S,Vbuild_walls) &
    Vbuild_walls >= 1 & Vbuild_walls <=1) &
   (mission_accomplished(S,fit_doors) | mplayers(fit_doors,S,Vfit_doors) &
    Vfit_doors >= 1 & Vfit_doors <= 1) &
   (mission_accomplished(S,fit_windows) | mplayers(fit_windows,S,Vfit_windows) &
    Vfit_windows >= 1 & Vfit_windows <=1) &
   (mission_accomplished(S,management_of_house_building) |
    mplayers(management_of_house_building,S,Vmanagement_of_house_building) &
    Vmanagement_of_house_building >= 1 & Vmanagement_of_house_building <= 1) &
   (mission_accomplished(S,build_roof) | mplayers(build_roof,S,Vbuild_roof) &
    Vbuild_roof >= 1 & Vbuild_roof <= 1) &
   (mission_accomplished(S,paint_house) | mplayers(paint_house,S,Vpaint_house) &
    Vpaint_house >= 1 & Vpaint_house <=1) &
   (mission_accomplished(S,install_electrical_system) |
    mplayers(install_electrical_system,S,Vinstall_electrical_system) &
    Vinstall_electrical_system >= 1 & Vinstall_electrical_system <= 1) &
   (mission_accomplished(S,prepare_site) | mplayers(prepare_site,S,Vprepare_site) &
    Vprepare_site >= 1 & Vprepare_site <= 1) &
   (mission_accomplished(S,lay_floors) | mplayers(lay_floors,S,Vlay_floors) &
    Vlay_floors >= 1 & Vlay_floors <= 1) &
   (mission_accomplished(S,install_plumbing) |
    mplayers(install_plumbing,S,Vinstall_plumbing) &
    Vinstall_plumbing >= 1 & Vinstall_plumbing <= 1).
is_finished(S) :- satisfied(S,house_built).
mission_accomplished(S,M) :- .findall(Goal, mission_goal(M,Goal), MissionGoals) &
                             all_satisfied(S,MissionGoals).
all_satisfied(_,[]).
all_satisfied(S,[G|T]) :- satisfied(S,G) & all_satisfied(S,T).
any_satisfied(S,[G|_]) :- satisfied(S,G).
any_satisfied(S,[_|T]) :- any_satisfied(S,T).

// enabled goals (i.e. dependence between goals)
enabled(S,G) :- goal(_, G,  dep(or,PCG), _, NP, _) & NP ≡ 0 & any_satisfied(S,PCG).
enabled(S,G) :- goal(_, G, dep(and,PCG), _, NP, _) & NP ≡ 0 & all_satisfied(S,PCG).
super_satisfied(S,G) :- super_goal(SG,G) & satisfied(S,SG).

// ** Norms

norm n1:
        scheme_id(S) & responsible(Gr,S) &
        mplayers(management_of_house_building,S,V) & V < 1 &
        fplay(A,house_owner,Gr) &
        // if all mission's goals are satisfied,
        // the agent is not obliged to commit to the mission
        not mission_accomplished(S,management_of_house_building)
```

```
      -> obligation(A,n1,committed(A,management_of_house_building,S), `now`+`2 minutes`).
norm n2:
        scheme_id(S) & responsible(Gr,S) &
        mplayers(prepare_site,S,V) & V < 1 &
        fplay(A,site_prep_contractor,Gr) &
        not mission_accomplished(S,prepare_site)
     -> obligation(A,n2,committed(A,prepare_site,S), `now`+`1 year`).
norm n3:
        scheme_id(S) & responsible(Gr,S) &
        mplayers(lay_floors,S,V) & V < 1 &
        fplay(A,bricklayer,Gr) &
        not mission_accomplished(S,lay_floors)
     -> obligation(A,n3,committed(A,lay_floors,S), `now`+`1 year`).
norm n4:
        scheme_id(S) & responsible(Gr,S) &
        mplayers(build_walls,S,V) & V < 1 &
        fplay(A,bricklayer,Gr) &
        not mission_accomplished(S,build_walls)
     -> obligation(A,n4,committed(A,build_walls,S), `now`+`1 year`).
norm n5:
        scheme_id(S) & responsible(Gr,S) &
        mplayers(build_roof,S,V) & V < 1 &
        fplay(A,roofer,Gr) &
        not mission_accomplished(S,build_roof)
     -> obligation(A,n5,committed(A,build_roof,S), `now`+`1 year`).
norm n6:
        scheme_id(S) & responsible(Gr,S) &
        mplayers(fit_windows,S,V) & V < 1 &
        fplay(A,window_fitter,Gr) &
        not mission_accomplished(S,fit_windows)
     -> obligation(A,n6,committed(A,fit_windows,S), `now`+`1 year`).
norm n7:
        scheme_id(S) & responsible(Gr,S) &
        mplayers(fit_doors,S,V) & V < 1 &
        fplay(A,door_fitter,Gr) &
        not mission_accomplished(S,fit_doors)
     -> obligation(A,n7,committed(A,fit_doors,S), `now`+`1 year`).
norm n8:
        scheme_id(S) & responsible(Gr,S) &
        mplayers(install_plumbing,S,V) & V < 1 &
        fplay(A,plumber,Gr) &
        not mission_accomplished(S,install_plumbing)
     -> obligation(A,n8,committed(A,install_plumbing,S), `now`+`1 year`).
norm n9:
        scheme_id(S) & responsible(Gr,S) &
        mplayers(install_electrical_system,S,V) & V < 1 &
        fplay(A,electrician,Gr) &
        not mission_accomplished(S,install_electrical_system)
     -> obligation(A,n9,committed(A,install_electrical_system,S), `now`+`1 year`).
norm n10:
        scheme_id(S) & responsible(Gr,S) &
        mplayers(paint_house,S,V) & V < 1 &
        fplay(A,painter,Gr) &
        not mission_accomplished(S,paint_house)
     -> obligation(A,n10,committed(A,paint_house,S), `now`+`1 year`).

// --- Goals ---
// agents are obliged to fulfill their enabled goals
norm ngoal:
        committed(A,M,S) & mission_goal(M,G) & goal(_,G,_,achievement,_,D) &
        well_formed(S) & not satisfied(S,G) & enabled(S,G) &
```

```
          not super_satisfied(S,G)
       -> obligation(A,ngoal(S,M,G),achieved(S,G,A),`now` + D).

  // --- Properties check ---
  norm goal_non_compliance:
          obligation(Agt,ngoal(S,M,G),Obj,TTF) &
          not Obj &
          `now` > TTF
       -> fail(goal_non_compliance(obligation(Agt,ngoal(S,M,G),Obj,TTF))).
  norm mission_permission:
          committed(Agt,M,S) &
          not (mission_role(M,R) &
          responsible(Gr,S) &
          fplay(Agt,R,Gr))
       -> fail(mission_permission(Agt,M,S)).
  norm mission_left:
          leaved_mission(Agt,M,S) &
          not mission_accomplished(S,M)
       -> fail(mission_left(Agt,M,S)).
  norm mission_cardinality:
          scheme_id(S) &
          mission_cardinality(M,_,MMax) &
          mplayers(M,S,MP) &
          MP > MMax
       -> fail(mission_cardinality(M,S,MP,MMax)).
  norm ach_not_enabled_goal:
          achieved(S,G,Agt) &
          mission_goal(M,G) &
          not mission_accomplished(S,M) &
          not enabled(S,G)
       -> fail(ach_not_enabled_goal(S,G,Agt)).
  norm ach_not_committed_goal:
          achieved(S,G,Agt) &
          .findall(M, mission_goal(M,G) &
          (committed(Agt,M,S) | mission_accomplished(S,M)), [])
       -> fail(ach_not_committed_goal(S,G,Agt)).
// end of scheme build_house_sch
```

APÊNDICE B – Constitutive specification of the *Build-a-house* example

```
status_functions:
 states: play(A,R,G), responsible(G,S), committed(A,Mission,S), achieved(S,G,A).

constitutive_rules:
  /* In this application, the agent Giacomo is allways the house owner. */
  1: count-as play(giacomo,house_owner,"hsh_group").

  /* Rules 2 to 10: The state where the property currentWinner(Auction,Agent)
     holds counts as Agent playing some role in the house building.*/
  2: currentWinner(auction_for_SitePreparation,Agent)
     count-as play(Agent,site_prep_contractor,"hsh_group")
     while nticks(clock,Time)&(Time>5000).

  3: currentWinner(auction_for_Floors,Agent)
     count-as play(Agent,bricklayer,"hsh_group")
     while nticks(clock,Time)&(Time>5000).

  4: currentWinner(auction_for_Walls,Agent)
     count-as play(Agent,bricklayer,"hsh_group")
     while nticks(clock,Time)&(Time>5000).

  5: currentWinner(auction_for_Roof,Agent)
     count-as play(Agent,roofer,"hsh_group")
     while nticks(clock,Time)&(Time>5000).

  6: currentWinner(auction_for_WindowsDoors,Agent)
     count-as play(Agent,window_fitter,"hsh_group")
     while nticks(clock,Time)&(Time>5000).

  7: currentWinner(auction_for_WindowsDoors,Agent)
     count-as play(Agent,door_fitter,"hsh_group")
     while nticks(clock,Time)&(Time>5000).

  8: currentWinner(auction_for_Plumbing,Agent)
     count-as play(Agent,plumber,"hsh_group")
     while nticks(clock,Time)&(Time>5000).

  9: currentWinner(auction_for_ElectricalSystem,Agent)
     count-as play(Agent,electrician,"hsh_group")
     while nticks(clock,Time)&(Time>5000).

 10: currentWinner(auction_for_Painting,Agent)
     count-as play(Agent,painter,"hsh_group")
     while nticks(clock,Time)&(Time>5000).

 /* Defining what counts as, from the instituional perspective, a well formed group.
    This constitution is necessary to constitute the commitments */
 11: count-as responsible("hsh_group","bhsch")
     while play(Electrician1,electrician,"hsh_group")  &
           not(play(Electrician2,electrician,"hsh_group") &
               not(Electrician1==Electrician2)) &
           play(Site_prep_contractor1,site_prep_contractor,"hsh_group") &
           not(play(Site_prep_contractor2,site_prep_contractor,"hsh_group") &
               not(Site_prep_contractor1==Site_prep_contractor2))&
           play(Bricklayer1,bricklayer,"hsh_group") &
           play(Bricklayer2,bricklayer,"hsh_group") &
           not(play(Bricklayer3,bricklayer,"hsh_group") &
               not(Bricklayer1==Bricklayer2) & not(Bricklayer2==Bricklayer3))&
           play(Plumber1,plumber,"hsh_group")  &
           not(play(Plumber2,plumber,"hsh_group") &
               not(Plumber1==Plumber2)) &
           play(Window_fitter1,window_fitter,"hsh_group") &
           not(play(Window_fitter2,window_fitter,"hsh_group")&
               not(Window_fitter1==Window_fitter2))&
           play(Door_fitter1,door_fitter,"hsh_group")&
            not(play(Door_fitter2,door_fitter,"hsh_group") &
               not(Door_fitter1==Door_fitter2)) &
           play(Roofer1,roofer,"hsh_group") &
```

```
          not(play(Roofer2,roofer,"hsh_group") & not(Roofer1==Roofer2))&
          play(House_owner1,house_owner,"hsh_group") &
          not(play(House_owner2,house_owner,"hsh_group") &
              not(House_owner1==House_owner2)).


/* Rules 13 to 22 (2nd order constitution): An the state of an agent A
   carrying the status function Y counts as the state committed(A,M,S) in the institution */
12: play(A,house_owner,"hsh_group")
    count-as committed(A,management_of_house_building,"bhsch")
    while responsible("hsh_group","bhsch").

13: play(A,site_prep_contractor,"hsh_group")
    count-as committed(A,prepare_site,"bhsch")
    while responsible("hsh_group","bhsch").

14: play(A,bricklayer,"hsh_group")
    count-as committed(A,lay_floors,"bhsch")
    while responsible("hsh_group","bhsch").

15: play(A,bricklayer,"hsh_group")
    count-as committed(A,build_walls,"bhsch")
    while responsible("hsh_group","bhsch").

16: play(A,roofer,"hsh_group")
    count-as committed(A,build_roof,"bhsch")
    while responsible("hsh_group","bhsch").

17: play(A,window_fitter,"hsh_group")
    count-as committed(A,fit_windows,"bhsch")
    while responsible("hsh_group","bhsch").

18: play(A,door_fitter,"hsh_group")
    count-as committed(A,fit_doors,"bhsch")
    while responsible("hsh_group","bhsch").

19: play(A,plumber,"hsh_group")
    count-as committed(A,install_plumbing,"bhsch")
    while responsible("hsh_group","bhsch").

20: play(A,electrician,"hsh_group")
    count-as committed(A,install_electrical_system,"bhsch")
    while responsible("hsh_group","bhsch").

21: play(A,painter,"hsh_group")
    count-as committed(A,paint_house,"bhsch")
    while responsible("hsh_group","bhsch").




/* Rules 23 to 22: the occurrence of some events in the environment counts-as,
    in the institution, the state achieved(S,G,A) */
22: count-as achieved("bhsch",site_prepared,Agent)
    when prepareSite[sai__agent(Agent)].

23: count-as achieved("bhsch",electrical_system_installed,Agent)
    when installElectricalSystem[sai__agent(Agent)].

24: count-as achieved("bhsch",floors_laid,Agent)
    when layFloors[sai__agent(Agent)].

25: count-as achieved("bhsch",walls_built,Agent)
    when buildWalls[sai__agent(Agent)].

26: count-as achieved("bhsch",roof_built,Agent)
    when buildRoof[sai__agent(Agent)].

27: count-as achieved("bhsch",windows_fitted,Agent)
    when fitWindows[sai__agent(Agent)].
```

```
28: count-as achieved("bhsch",doors_fitted,Agent)
    when fitDoors[sai__agent(Agent)].

29: count-as achieved("bhsch",plumbing_installed,Agent)
    when installPlumbing[sai__agent(Agent)].

30: count-as achieved("bhsch",electrical_system_installed,Agent)
    when installElectricalSystem[sai__agent(Agent)].

31: count-as achieved("bhsch",exterior_painted,Agent)
    when paintExterior[sai__agent(Agent)].

32: count-as achieved("bhsch",interior_painted,Agent)
    when paintInterior[sai__agent(Agent)].
```