



HAL
open science

Situated artificial institutions: stability, consistency, and flexibility in the regulation of agent societies

Maiquel de Brito, Jomi Fred Hübner, Olivier Boissier

► To cite this version:

Maiquel de Brito, Jomi Fred Hübner, Olivier Boissier. Situated artificial institutions: stability, consistency, and flexibility in the regulation of agent societies. *Autonomous Agents and Multi-Agent Systems*, 2018, 32 (2), pp.219-251. 10.1007/s10458-017-9379-3 . hal-01988011

HAL Id: hal-01988011

<https://hal.science/hal-01988011v1>

Submitted on 16 Nov 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - NonCommercial 4.0 International License

Situated artificial institutions: stability, consistency, and flexibility in the regulation of agent societies

Maiquel de Brito¹  · Jomi Fred Hübner² · Olivier Boissier³

Abstract In multi-agent systems, norms are a usual way to regulate the behaviour of autonomous agents. To be stable in different circumstances, norms are specified using high level terms, abstracting from the particular dynamics of the environment where the agents are situated. However, applying these norms requires a proper link with a concrete environment. Detaching that link from the norms themselves provides stability to the normative regulation but raises consistency and flexibility issues. Consistency is achieved when the abstract norms are coherent with the environment under regulation. Flexibility is achieved when different kinds of norms share the same interpretation about the environmental state. These properties are provided in some current works. However, since they are interrelated, there is not, to our knowledge, a single proposal providing all of them. This paper proposes the situated artificial institution (SAI) model to address these three issues—stability, consistency, and flexibility—by conceiving norms as part of institutions that provide, through the process of constitution, a social interpretation of the environmental state. After the presentation of the formalised model of SAI, a case study is used to illustrate and test this approach.

Keywords Institutions · Norms · Count-as · Situatedness

✉ Maiquel de Brito
maiquel.brito@rolante.ifrs.edu.br

Jomi Fred Hübner
jomi.hubner@ufsc.br

Olivier Boissier
Olivier.Boissier@emse.fr

¹ Federal Institute of Education, Science and Technology of Rio Grande do Sul, Rolante, Rio Grande do Sul, Brazil

² UFSC/CTC/DAS/PPGEAS, Federal University of Santa Catarina, PO Box 476, Florianópolis, SC 88040-900, Brazil

³ Laboratoire Hubert Curien UMR CNRS 5516, Institut Henri Fayol, MINES Saint-Etienne, Saint-Étienne, France

1 Introduction

This paper focuses on open software systems where autonomous agents can enter in, act to achieve their goals, and eventually exit from. Some examples of systems that can be modelled as open multi-agent systems are the *Wikipedia* and their *bots* [40,41], financial markets and their algorithm traders [37], smartgrids [56], collaborative platforms for crisis management [70], and e-commerce systems [3,22]. These open multi-agent systems (MAS) may be seen as *agent societies* as their participants usually have some social behaviour [67]. For instance, in e-commerce, agents may compete, negotiate, build coalitions, etc. These societies require some regulation of the behaviour of the autonomous agents to conciliate their individual goals and social expectations [19,57]. Regulative norms (henceforth referred just as *norms*) based on concepts such as obligations, prohibitions, permissions, duties, rights, authorisations, powers, etc., are a usual way to express the expected agents' behaviour in the society [11].

Independent of any regulation, the arena of the behaviour of the agents is, by definition, the *environment*, that is the set of elements that are perceived and acted upon by the agents to achieve their goals (e.g. sensors and actuators, printers, networks, databases, web services, etc.) [39,62,64,77,78]. Environments are typically dynamic, changing along the execution of the system [46,64]. Such changes may be the results of the actions realised by the agents on the elements composing the environment but can also be the result of the internal dynamics of these elements. Changes in the environment are usually beyond the control of the norms and may be even unpredictable when norms are designed.

Designing norms to regulate the activities of the agents in such a dynamic environment tends to be complex due to factors such as the unpredictable behaviour of the environmental elements or the evolving way the systems' expectations are concretely realised. For example, in an e-commerce scenario where the agents are expected to pay for their purchases, both the set of agents that should pay and the means to perform payments may change along the execution of the system. The norm "bob is obliged to make a bank deposit" should be changed to cover additional agents and payment methods (e.g. "bob and tom are obliged either to make a bank deposit or to use a credit card").¹ A usual solution to make the norm more stable is to abstract from the concrete environment, defining it in terms of "buyer" and "payment" instead of referring to each possible agent name and payment method [1,44]. But applying norms designed in such an abstract way is difficult as the practical applications are not composed of elements such as buyers and payments. Rather, they are composed of agents, messages, webservices, etc., placed in the environment, that in certain contexts may *count as* buyers and payments. To be applied, the norms must be *situated* in the environment. That is to say, the abstract concepts used in the norms (e.g. "buyer", "payment") must be *coupled* with the concrete environmental elements (e.g. "bob", "bank deposit") involved in the regulation. This requirement rises two additional complexities. First, if the nature of buyer and payment is not explicitly defined, the normative system cannot assume that they refer, respectively, to agents and to their actions. For example, one could couple payment to an agent and buyer to an action, loosing *consistency* between the norm and the environment. Second, such coupling should be shared by different sets of norms—possibly expressed in different normative models and languages—regulating the same system (e.g. norms from the store, financial, and logistic systems in an e-commerce scenario). Otherwise, different sets of norms might prescribe conflicting behaviours. For instance, financial norms could consider

¹ This paper does not have in mind just e-commerce systems. The focus is on multi-agent systems. The e-commerce scenario is used to illustrate this kind of system.

an agent as *buyer*, obliging him to pay for the purchases, while logistic norms do not consider the agent as *buyer*, forbidding him to receive the purchased goods.

From this example, we can see three important but challenging properties of normative systems: *stability*, *consistency*, and *flexibility*. Although the words *stability*, *consistency*, and *flexibility* are used in the literature to refer to different notions, they have a particular denotation in this paper. *Stability* is the property of a normative system to range over different environmental circumstances, even unknown when the norms are specified. Systems having stability can support changes in the environment under regulation without changing the norms (norms change only when the regulative requirements change). *Consistency* is the property of a normative system to *correctly* couple terms in the norms with environment elements of different natures, distinguishing, for instance, agents from actions. Systems having consistency ensure that norms are coherent with the environment under regulation, even abstracting from it. *Flexibility* is the property of a normative system to include different kinds of norms to regulate the same system. Systems having flexibility support the regulation provided by different normative models keeping the same coupling of the normative layer to the environment. Stability is important to regulate systems where not only the agents, but also the environmental elements supporting their actions, can change and can be even unknown when the norms are designed. Consistency is important as the regulation is expected to be enforced matching with the actual situations in the environment where the agents act. Flexibility is important as different regulative requirements, that can also change along the system execution, can be better expressed by different normative models. These properties are provided in some current works. However, as shown in Sect. 2, there is not, to our knowledge, a single proposal providing all of them.

This paper aims to investigate how to provide stability, consistency, and flexibility to the regulation of MAS. Our main inspiration comes from works in the field of the social sciences that conclude that, in human societies, the regulation of their participants' expected behaviour is inserted in an *institutional reality* [65,66]. Such institutional reality makes the regulation of human societies stable, consistent, and flexible even with people acting in a dynamic world. The result of this work is a model of *artificial institution* called situated artificial institutions² (SAI), that, being introduced in [23–25], is extended in this paper. SAI considers norms as part of *institutions* that contain also the elements to represent the institutional reality and to connect norms in it, providing stability, consistency, and flexibility to the regulation in MAS. In this paper we present our contribution conceiving *constitution*—the process that builds the institutional reality—as a key element of SAI, presenting its particular representations and dynamics. The constitution produces *status function assignments*, that are the building blocks of the proposed representation of institutional reality, consisting of the institutional interpretation of the environmental elements. This is an important difference compared to the current state of the art where some works consider institutional reality as being the institutional vocabulary, i.e. the set of words used to specify the norms [1,12], while other works consider it as composed of facts related to the dynamics of the norms (e.g. some event counts as a norm fulfilment) [32,33,61], and others consider it as being *institutional events*, that are an institutional counterpart of events occurring in the environment [17,20,38,74,75].

This paper is organised as follows: Sect. 2 discusses the challenges of stability, consistency, and flexibility in the regulation of MAS, describing some existing ways to tackle them; Sect. 3 presents constitution as conceived in SAI; Sect. 4 describes how norms can be coupled in the institutional reality produced by such constitution. Stability, consistency, and flexibility

² The notion of *Artificial Institution* is proposed in [38] to refer to an extension to the concept of *electronic institution* [36]. While electronic institutions focus on norms, artificial institutions contain additional elements to represent other social aspects of MAS. The social aspect focused in our work is the *institutional reality*.

provided by such coupling are analysed in Sect. 5. Section 6 discuss some limitations of the proposed model and of this paper, motivating future work. Section 7 exposes some conclusions and perspectives.

2 Stability, consistency, and flexibility of normative regulation: challenges and state of the art

Achieving stability, consistency, and flexibility are interrelated challenges. The most usual approach to achieve norm stability is to specify the norms in an abstract level and then to link them to the environment [1,43]. But this solution raises issues of consistency and flexibility. Sections 2.1–2.3 explain how stability, consistency, and flexibility are provided by the existing approaches and how these issues are interrelated.

2.1 Stability

In MAS, norms regulate interactions in a dynamic environment, whose elements can change along the time and can even be unknown when norms are designed. These features can bring instability to norms directly linked to the environment, as they need to be rewritten every time a new environmental element must be taken into account. To provide stability, current works usually consider that norms are specified in an abstract level instead of being directly related to the environment. They can thus range over different elements under regulation [1,43]. A norm stating that “buyers are obliged to pay” can range over all the agents that could be considered as *buyer* and over all the actions considered as *payment*, independent of their concrete realisation in the environment. The norm is thus stable as it does not require changes when the set of agents considered as *buyer* and the set of possible payment methods change.

The effectiveness of norms specified in such an abstract way, however, requires a link between the abstract concepts and the environment where the agents act. This link is usually represented through *count-as rules* that define that facts occurring in the environment *count as* facts within the institutions [42,65]. The existing approaches on *count as* can be classified into different categories according to Grossi and Jones [45]. This section focuses on works that can be classified as addressing *count-as related to regulative norms*, i.e. those works concerned with relating the normative regulation to some interpretation of the environment produced by means of count-as. For instance, Aldewereld et al. [1] and Boella and van der Torre [8–10,12] propose count-as rules to link the abstract concepts used in the norms to the environmental elements defining, for example, that the agent *bob* counts as a *buyer* and the action *credit_op*(100) counts as a *payment*. Instead of linking the concepts used in the norms to the environment, Dastani et al. [32,33] propose count-as rules to define the environmental conditions that trigger changes in the state of the norm instances (i.e. environmental facts count as violations, fulfilments, etc. of the norms). For example, one could specify that “*credit_op*(100)” counts as the *fulfilment* of the norm “*buyers are obliged to pay*” instead of counting as a *payment*. In a similar direction, Piunti et al. [61] consider that environmental events count as changes in the state of *MOISE* organisations [49,50].

In this approach based on count-as, the stability problem is indeed moved from the norm to the count-as specifications. Changes in the environment are covered by the count-as rules without requiring changes in the norms. The count-as rules and their management resources have special features to handle this issue, leaving to the norms the concern of modelling the regulative requirements of the system.

2.2 Consistency

By abstracting from the environment, norms are decoupled from the actual nature and dynamics of the environmental elements involved in the regulation. Norms can turn inconsistent if these nature and dynamics are not taken into account by the count-as approach. Aldewereld et al. [1] and Boella and van der Torre [8–10, 12] do not consider how the nature of the different environmental elements is related with the different concepts used in the norms. In the aforementioned norm, it is not possible to ensure neither that *buyer* will refer to an agent nor that *payment* will refer to an action. One could wrongly specify a count-as rule that *payment* points to an agent and *buyer* points to an action. As the nature of the environmental elements is ignored, the dynamics associated to such nature is also ignored. For example, agents can enter and leave the system and, even if *buyer* is properly linked to an agent, these works do not consider explicitly how to manage situations where agents considered as buyers leave the system. Obligations could stand to *buyers* that are no longer participating to the system.

Linking environmental facts to the dynamics of the norms [32, 33, 61] may also 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 norms “buyer is obliged to pay while the store is open” and “buyer is prohibited to pay while the store is not open” even if the norms are declared as being active under opposite conditions. The state “store is open” is not linked to the environment and thus these opposite conditions cannot be checked with respect to the actual environment.

Works providing consistency consider some correspondence between the abstract concepts used in the norms and the nature of the environmental elements they refer to. Fornara et al. [38], Viganò and Colombetti [74, 75], Cardoso and Oliveira [17], and Cliffe et al. [20] consider that events occurring in the environment can count as *institutional events*, that have a precise semantics. It provides *norm consistency* as if some element of a norm is an institutional event, then it is ensured that it points to an event in the environment. But, as noted by Vos et al. [76], regulation of MAS should be based on events and also on states, which are not supported by these works. There is not, in this case, *norm stability* with respect to the environmental states.

2.3 Flexibility

Norms can express the expected agents’ behaviour in many different ways.³ For example, some normative models express norms only as obligations [49] while others consider also prohibitions [72] and permissions [59]; or, for instance, some models consider that the agents can repair their misbehaviour [60] while others consider that they are sanctioned [72].

Since different normative models can take part in the same institution [21], all the norms following the different models should be uniformly linked to the same environment. For example, every norm referring to *buyer* and *payment* should link these concepts to the environmental elements in the same way. Providing such uniform link between norms and environment, making the regulation flexible to support different normative models, is challenging considering the diversity of normative models and their conceptions of regulation.

Although the works by Aldewereld et al. [1], Boella and van der Torre [8–10, 12], Fornara et al. [38], Viganò and Colombetti [74, 75], Cliffe et al. [20], and Cardoso and Oliveira [17] consider specific normative models, their general idea of linking the concepts used in the norms to the environment could be applied to different normative models. Using generic

³ Among the huge literature in the field, the interested reader can find detailed information on norms in [2, 7, 13] and in the COIN series of workshops (<http://www.pcs.usp.br/~coin/>).

Table 1 Synthesis of the analysis of related works with respect to stability, consistency and flexibility of the normative regulation, where “✓” means *achieved* and “–” means *not achieved*

	Stability	Consistency	Flexibility
– Aldewereld et al. [1]	✓	–	✓
– Boella and van der Torre [8–10,12]	✓	–	✓
– Fornara et al. [38]	–	✓	✓
– Viganò and Colombetti [74,75]	–	✓	✓
– Cliffe et al. [20]	–	✓	✓
– Cardoso and Oliveira [17]	–	✓	✓
– Dastani et al. [32,33]	✓	–	–
– Campos et al. [16]	–	–	✓
– Piunti et al. [61]	✓	–	–
– Brito et al. [26]	–	–	✓

interfaces to inspect the environment and to change the normative state accordingly, as proposed in [16,26], can provide flexibility as interfaces can be flexible enough to link different normative models to the same environment. But stability and consistency depend on the way the interfaces change the state of each normative model regulating the system and thus cannot be assured. Linking the environmental facts to the dynamics of specific models, as proposed in [32,33,61], limits such flexibility as the specification of the relation between environmental facts and normative dynamics cannot be applied to other kinds of norms in the same system.

2.4 Synthesis of the state of the art

As we can see in Table 1, there is not an approach providing stability, consistency, and flexibility altogether to the regulation in MAS. The best that current works do is to conciliate flexibility either with stability or with consistency. Works providing both flexibility and stability consider that the concepts used in the norms are linked to the environmental elements without taking into account the nature of the elements to be linked, falling in inconsistencies. On the other hand, works providing both flexibility and consistency consider that the concepts used in the norms are of a specific “type”, i.e. *institutional events*, that can be only linked to environmental events, falling in instability as elements other than events must be directly referred in the norms.

We aim to fill this gap to be able to design MAS where the regulation—independent of how it is expressed—is decoupled from, but still consistent with, the environmental elements involved in the achievement of the systems’ goals. To this end, we consider that norms are not the only element necessary to regulate MAS. Rather, they are part of *artificial institutions* (or simply *institutions*). Institutions enable the *institutional reality* of MAS, that is an interpretation of the environmental elements under regulation [6,53,63,65,66]. Norms are specified referring to concepts belonging to this institutional reality (being thus stable), that are endowed with a precise semantics to ensure consistency. Flexibility is achieved as the elements belonging to the institutional reality are supposed to be accessible to several kinds of norms.

Thus, our proposal to provide stability, consistency and flexibility to the regulation of MAS is not just to design norms, but to design *institutions* (that comprise norms). The next

section explains our conception of *constitution*, that is the process that builds the institutional reality. Section 4 explains how norms fit in this institutional reality. Considering the several normative models already present in the literature, we focus the paper on the constitution part of the institution, which is the our main contribution, and simply couple it with existing normative models to highlight the flexibility.

3 Constitution: representing and building the institutional reality in SAI

In SAI, *constitution* is the process of building the institutional reality that norms are based on.⁴ Section 3.1 presents the philosophical background of this conception. Sections 3.2–3.5 introduce *constitution*. Section 3.6 illustrates the evolution of the constitutive state, that is the representation of the institutional reality in SAI, based on the environmental dynamics.

3.1 Philosophical background

According to Searle [65,66], the social reality where human people are immersed arises from the concrete world (i.e. the environment) based on some elements including *deontic powers*, *status functions*, and *constitutive rules*. Deontic powers are the rights, duties, obligations, permissions, authorisations, etc. that direct the expected behaviour of the people in the society. Status functions are functions that environmental elements perform independent of their physical virtues. Constitutive rules specify the assignment of status functions to environmental elements. Constitutive rules have the form *X counts as Y in C* meaning that the physical element *X* has the status function *Y* in the context *C*. For example, a constitutive rule can define that a small line of stones (*X*) *counts as* the boundary of a private property (*Y*) in any circumstance (*C*). Due to such assignment of status function, people have reasons to follow the norm that states that they are forbidden to get into the private property even though they are physically able to cross the line of stones.

Searle claims that this system of constituted status functions is “the glue that holds society together” [66, p. 9] as it leads people to act according to the social expectations independent (1) of their own desires and inclinations and (2) of the physical virtues of the physical elements (e.g. a line of stones does not need to be the highest one to be a boundary, a man does not need to be the strongest one to be the leader, etc.).

3.2 SAI overview

This paper uses the notion of *norm* to denote the *deontic powers* referred by Searle because it is the usual way to represent the expected agents’ behaviour in MAS.⁵ Norms specify the expected behaviour from the agents in an abstract level that is not directly related 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 the institutional specification of an auction. The norm, however, does not specify aspects such as (1) what an agent should do to become the *winner* of the auction, (2) what an agent must do to perform the *payment*, or (3) how the *fine* is applied. In this scenario, *winner*, *payment*, and *fine* are all *status functions*: they are status, assigned

⁴ Norms refer to (and somehow depend) but are not part of the institutional reality [17,65,66]. The elements composing the institutional reality, as well as the norms, are part of a *social reality* [38,65,66], whose analysis is beyond the scope of this paper.

⁵ Basing other possible representations of deontic powers on SAI is a future work.

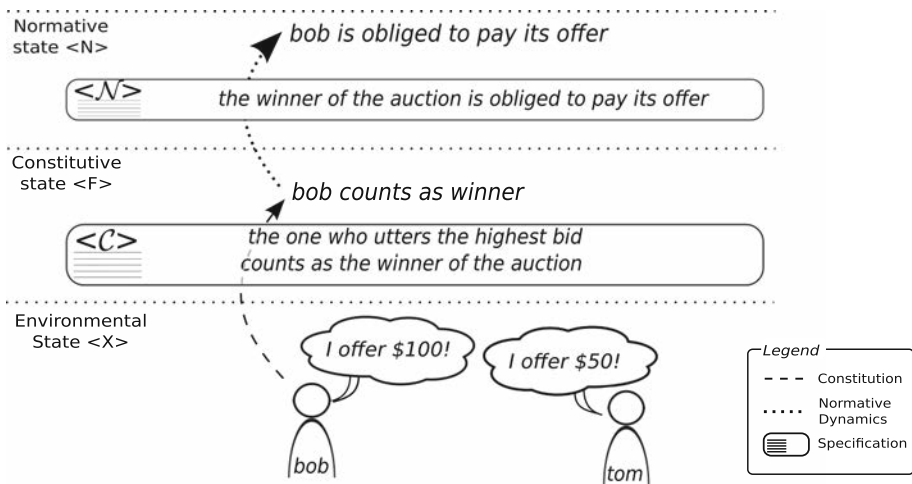


Fig. 1 SAI overview: constitutive rules specify how the constitutive state is built from the environmental state while norms specify how the normative state is built from the constitutive state

by the institution to the environmental elements, that impose functions to these elements.⁶ SAI considers that status functions are assigned to agents acting, events occurring, and states holding in the environment. For example, in a given institution:

- An agent may get the function of auctioneer. But it has such 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 \$10,000” may get 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 at 10 a.m.” may mean, in the institution, the minimum quorum for an auction.

SAI considers these three kinds of status functions because they are enough to cover the environmental elements involved in the normative regulation: agents are the bearers of obligations, prohibitions, etc. while the dynamics of the norms (activations, fulfilments, violations, etc.) is based on events and states, depending on the normative model [76].

The effectiveness of norms specified through status functions depends on their connection to the environment as their 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* (Fig. 1). The set of constituted status functions is the *constitutive state* of the institution, that can be seen as the institutional interpretation of the current environmental state. 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

⁶ In SAI, as in Searles’ work, the expression “status function” means 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.

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 (Fig. 1).

In the following, Sect. 3.3 describes how the constitution of status functions from the environmental elements is specified in SAI through constitutive rules. Section 3.4 describes the dynamics of such constitution, that builds the constitutive state, that is the representation of the institutional reality in SAI. The evolving of the normative state based on the constitutive one is addressed in Sect. 4.

3.3 Constitutive specification

The constitutive specification designs the institutional reality in SAI. It defines, through constitutive rules, what are the elements composing the institutional reality (i.e. what are the constituted status functions) according to the different possible environmental circumstances. In the following, Sect. 3.3.1 formally presents the elements involved in the constitutive specification while Sect. 3.3.2 proposes a language to specify constitutive rules based on the presented formalism.

3.3.1 Constitutive specification formalism

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). The presented formalism employs the following elements of the first-order logic language: (1) *constants*, that refer to elements existing in the modelled world; (2) atomic formulae (or simply *atoms*), that represent propositions; and *substitutions* of variables [14].⁷

Definition 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.

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 atom identifying the event [e.g. *offer*(100)] and a is (1) either a constant identifying the agent that has triggered the event or (2) ε if the event is produced by the environment itself (e.g. a clock tick). Properties in $\mathcal{S}_{\mathcal{X}}$ are represented by atoms.

It is important to observe that the set \mathcal{X} is just a *representation* of the elements that potentially take part to the environment.⁸ 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 2 (*Status function*) Status functions are functions that the environmental elements may perform in the institution independent of their design aspects. The set \mathcal{F} contains all the possible status functions of a SAI and is defined as $\mathcal{F} = \mathcal{A}_{\mathcal{F}} \cup \mathcal{E}_{\mathcal{F}} \cup \mathcal{S}_{\mathcal{F}}$ where $\mathcal{A}_{\mathcal{F}}$ is the

⁷ In this paper, a *substitution* is always represented by θ . A *substitution* is a finite set of pairs $\{\alpha_1/\beta_1, \dots, \alpha_n/\beta_n\}$ 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 .

⁸ It is beyond of the scope of this paper to deal in detail with the environment. We just consider the elements of \mathcal{X} as existing outside the institution, being available thanks to reliable interfaces.

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.

Definition 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 $x \in \mathcal{F} \cup \mathcal{X} \cup \{\varepsilon\}$ counts as (i.e. x has the status function) $y \in \mathcal{F}$ when the event $t \in \mathcal{E}_{\mathcal{F}} \cup \mathcal{E}_{\mathcal{X}} \cup \{\varepsilon\}$ has happened and while the condition represented by m holds.⁹

While in Searle’s theory the context of constitutive rules is true when generic circumstances c hold, we consider that the context is true (1) when some event t has happened and (2) while certain conditions expressed by m hold. In the case of $t = \varepsilon \wedge m = \top$, 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 (cf. Definition 5). The process of establishing a SFA of y to some x , described in Sects. 3.4.2 and 3.4.3, is the *constitution* of y . The *constitution* of the status function y is performed as follows:

- *Assignment to an element x* This kind of constitution applies to constitutive rules where $x \neq \varepsilon$. In this case, the 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. The rule $\langle \text{bob}, \text{bidder}, (\text{offer}(10), \text{bob}), \text{auction_running} \rangle$ is an example of assignment of status function to a concrete element, that we name *first-order constitution* (cf. Sect. 3.4.2): it means that the agent *bob* carries the status function of *bidder* after having uttered its offer and while the auction is running. An example of assignment of status function to another status function, that we name *second-order constitution* (cf. Sect. 3.4.3) is the rule $\langle \text{bidder}, \text{auction_participant}, \varepsilon, \varepsilon \rangle$: it assigns the status function of *auction_participant* to the agents that carry the status function of *bidder*.
- *Freestanding assignment* This kind of constitution applies to rules where $x = \varepsilon$. 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 the rule $\langle \varepsilon, \text{auction_running}, \varepsilon, \neg \text{auction_finished} \rangle$ 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 recognised by Searle [65,66] and by other related authors [47,69].

3.3.2 Constitutive specification language

From the previously described elements, we introduce a language to specify constitutive rules. The constitutive specification, written based on the syntax given in Fig. 2, defines the sets of status functions (\mathcal{F}) and constitutive rules (\mathcal{C}) of the institution. Each constitutive rule (*const_rule* in the grammar of Fig. 2) has an identifier (*id*). Furthermore, the rules have the operator `count-as`, that performs the constitution of the status functions. The

⁹ ε represents that the element is not present in the constitutive rule.

```

const_model ::= F C
  F ::= "status_functions:" AF? EF? SF?
  AF ::= "agents:" aF(, aF) * .
  EF ::= "events:" eF(, eF) * .
  SF ::= "states:" sF(, sF) * .
  C ::= "constitutive_rules:" const_rule+
  const_rule ::= id : count_as_stat t? m_expression?.
count_as_stat ::= ((aF|aX|var) "count-as" aF) |
  ((eF|eX|var) "count-as" eF) |
  ((sF|sX|var)? "count-as" sF)
  t ::= "when" eF|eX
m_expression ::= "while" m
  m ::= sX|eX|sF|eF|not m|m "|" m|m "&" m|is_expr|"false"|"true"
is_expr ::= aX "is" aF|eX "is" eF|sX "is" sF
  aF ::= constant
  aX ::= constant
  eF ::= atom
  eX ::= atom(["sai__agent"(aX|var)])?
  sF ::= atom
  sX ::= atom
  id ::= constant

```

Fig. 2 Grammar of the constitutive specification language

elements related to the context of the constitutive rule (t and m) are optional. The element m in the grammar of Fig. 2 corresponds to the logical formulae used to check conditions for constitution, where "not", "|", "&", "false", and "true" correspond respectively to \neg , \vee , \wedge , \perp , and \top . The semantics of these formulae is given in page 1. The element a of an event $(e, a) \in \mathcal{E}_X$ is represented in the grammar of Fig. 2 by `sai__agent(a)`, that can be omitted to represent $a = \varepsilon$. Constants and atoms start by a lower case letter. Variables var are terms starting by an upper case letter.

Figure 3 shows the constitutive specification for the use case addressed in [29], where agents collaborate to manage crisis such as flooding, car crashes, etc. They act in an environment composed of geographic information systems (GIS) and of tangible tables [54] 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).

3.4 Constitutive dynamics

Constitutive elements (i.e. status functions assignments) are dynamically produced by the interpretation of the constitutive rules on the current state of the environment. The elements involved in this dynamics are introduced in Sect. 3.4.1. Then, Sects. 3.4.2 and 3.4.3 describe the process of constitution, that is the establishment of status function assignments.

```

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).

```

Fig. 3 Example of constitutive specification

3.4.1 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 are defined in this section.

Definition 4 (*Environmental state*) The actual environmental state is represented by $X = A_X \cup E_X \cup S_X$ where (1) A_X is the set of agents participating in the system, (2) E_X is the set of events occurring in the environment and (3) S_X is the set of environmental properties describing the environmental state.

Agents in A_X are represented by constants referring to their identifiers. 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, ε) .

Definition 5 (*Constitutive state*) The constitutive state of a SAI is the set of the existing SFAs. It is represented by $F = A_F \cup E_F \cup S_F$ where (1) $A_F \subseteq A_X \times \mathcal{A}_F$ is the set of agent-status function assignments, (2) $E_F \subseteq E_X \times \mathcal{E}_F \times A_X \cup \{\varepsilon\}$ is the set of event-status function assignments and (3) $S_F \subseteq S_X \cup \{\varepsilon\} \times \mathcal{S}_F$ is the set of state-status function assignments.

SFAs, established through the constitution of status functions as specified by constitutive rules (cf. Definition 3), are relations between environmental elements and status functions. Elements of A_F are pairs $\langle a_X, a_F \rangle$ meaning that the agent $a_X \in A_X$ has the status function $a_F \in \mathcal{A}_F$. Elements of E_F are triples $\langle e_X, e_F, a_X \rangle$ meaning that the event-status function $e_F \in \mathcal{E}_F$ is assigned to the event $e_X \in E_X$ produced by the agent $a_X \in A_X$. Events are usually considered at the individual agent level in normative systems [76]. Thus, it is

important to record the agent that causes an event-status function assignment to be able to check individually, in the normative level, the expected behaviour of the agents with respect to the production of event-status function assignments. Elements of S_F are pairs $\langle s_X, s_{\mathcal{F}} \rangle$ meaning that the state $s_X \in S_X$ carries the status function $s_{\mathcal{F}} \in S_{\mathcal{F}}$.

The constitution of status function is conditioned by the holding of the condition m (cf. Definition 3). Such condition is represented by the formulae m (hereafter *m-formulae*) defined in the grammar of Fig. 2. Considering a model $M = \langle F, X, \mathcal{F} \rangle$, the semantics of m-formulae is defined as follows:

$$\begin{aligned} M \models m \text{ iff } \exists \theta : & (m\theta \in E_X \vee m\theta \in S_X) \\ & \vee (\exists e_X : e_X \text{ is } m\theta) \\ & \vee (\exists s_X : s_X \text{ is } m\theta) \end{aligned} \quad (1)$$

$$\begin{aligned} M \models x \text{ is } y \text{ iff } \exists \theta : & (x\theta \in A_X \wedge y\theta \in \mathcal{A}_{\mathcal{F}} \wedge \langle x\theta, y\theta \rangle \in A_F) \\ & \vee (x\theta \in E_X \wedge x = (e, a) \wedge y\theta \in \mathcal{E}_{\mathcal{F}} \wedge \langle e\theta, y\theta, a\theta \rangle \in E_F) \\ & \vee (x\theta \in S_X \wedge y\theta \in S_{\mathcal{F}} \wedge \langle x\theta, y\theta \rangle \in S_F) \end{aligned} \quad (2)$$

Informally, from the expression 1, an m-formula m is true (1) if it represents either an event actually occurring or an state actually holding in the environment; (2) if it represents either an event-status function assigned to some environmental elements; or (3) if it represents a state-status function actually assigned to some environmental state. From the expression 2, an m-formula is true if it has the form $x \text{ is } y$ and either (1) x is an agent that carries the agent-status function y or (2) x is an event that actually carries the event-status function y or (3) x is a state that actually carries the state-status function y .

Definition 6 (*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 4 and 5. The normative state N depends on the dynamics of each normative model possibly composing the institution. Section 4 details how the state of norms following a specific normative model take part to SAI. Here, N subsumes the state of all the norms taking part in the institution.

Definition 7 (*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 $SAI_{Dyn}^i = \langle X^i, F^i, N^i \rangle$ where $X^i = A_X^i \cup E_X^i \cup S_X^i$ and $F^i = A_F^i \cup E_F^i \cup S_F^i$. N^i is the normative state at the step i (not detailed here as it depends on the different normative models taking part in the institution). A sequence of steps starting from the step s finishing in the step z (s.t. $s \in \mathbb{N}$, $z \in \mathbb{N}$, and $s < z$) is noted as $SAI_{Dyn}^s \dots SAI_{Dyn}^z$. Sequences of environmental, constitutive, and normative states are similarly noted, respectively, as $X^s \dots X^z$, $F^s \dots F^z$, and $N^s \dots N^z$.

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 4 illustrates these 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. 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

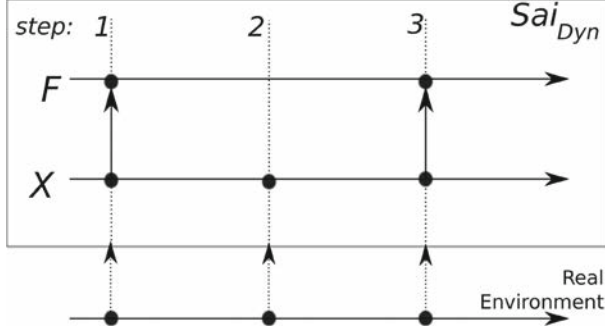


Fig. 4 SAI dynamics overview: changes in the real environment produce changes in the SAI representation of the environmental state that, in some cases, produce changes in the constitutive state

again, leading SAI to the state three, where changes are also produced in the constitutive state.

Constitutive rules can specify two kinds of constitution of status functions to produce SFAs: *first-order constitution* and *second-order constitution*. Sections 3.4.2 and 3.4.3 explain how SFA due to first- and second-order constitution can be deduced from a given SAI history.

3.4.2 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, Definitions 8–10 explain how this kind of constitution is produced. To formally define the constitutive dynamics, we use functions to define the set of SFAs that can be deduced from given environmental and constitutive states.

Definition 8 (*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:

$$\begin{aligned}
 f\text{-const}_a(\mathcal{F}, \mathcal{C}, X^0 \dots X^z, F^0 \dots F^z, i) = & \left\{ \langle x, y \rangle \mid x \in A_X^i \wedge y \in \mathcal{A}_{\mathcal{F}} \right. \\
 & \wedge \exists \langle x', y, t, m \rangle \in \mathcal{C} \exists \theta \exists s \in \mathbb{N} \forall k \in [s, i]: \\
 & (E_X^s \cup E_F^s \models t\theta) \wedge (X^k \cup F^k \models m\theta) \\
 & \left. \wedge x'\theta = x \right\} \quad (3)
 \end{aligned}$$

Informally, the function $f\text{-const}_a$ defines that (1) 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 (2) 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 . 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. $a \in A_X^i$ and $a \notin A_X^{i+1}$), 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 left the system

(cf. Sect. 3.5). The function also lays out our proposed approach to deal with combined instantaneous events and fluent states as conditions to constitution when it defines that an 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: (1) the repetition of the event t does not affect the SFA and (2) a SFA is dropped if m ceases to hold and is not activated again if the m turns to hold (unless the event t happens again while m is again holding).

The rule 1 in the Fig. 3 defines a first-order constitution of an agent-status function. If $checkin(table_mayor, bob) \in E_X^1$, meaning that the agent bob has checked in the $table_mayor$ at the step 1, then bob carries the status function $mayor$ [i.e. $\langle bob, mayor \rangle \in f\text{-const}_a(\mathcal{F}, \mathcal{C}, X^0 \dots X^z, F^0 \dots F^z, i)$] for all steps i , starting from the 1st one, while bob participates in the system (considering $\theta = \{Agent/bob\}$).

Definition 9 (*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:

$$f\text{-const}_s(\mathcal{F}, \mathcal{C}, X^0 \dots X^z, F^0 \dots F^z, i) = \left\{ \langle x, y \rangle \mid ((x = \varepsilon) \vee (x \in S_X^i)) \wedge (y \in \mathcal{S}_{\mathcal{F}}) \right. \\ \wedge \exists \langle x', y, t, m \rangle \in \mathcal{C} \exists \theta \exists s \in \mathbb{N} \forall k \in [s, i]: \\ \left. (E_X^s \cup E_F^s \models t\theta) \wedge (X^k \cup F^k \models m\theta) \wedge x'\theta = x \right\} \quad (4)$$

Similar to the constitution of agent-status functions, (1) a SFA is only assigned to a state $x \in S_X$ only if x actually holds in the environment and (2) 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 $f\text{-const}_s$ lays out our conception that the constitution of state-status functions may result in freestanding assignments.

The rule 6 in the Fig. 3 defines a first-order constitution of a state-status function. If $security_phase(downtown, emergency) \in E_X^3$, meaning that the GIS points the downtown as being insecure, 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 10 (*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:

$$f\text{-const}_e(\mathcal{F}, \mathcal{C}, X^0 \dots X^i, F^0 \dots F^i, i) = \left\{ \langle e, y, a \rangle \mid (e, a) \in E_X^i \wedge y \in \mathcal{E}_{\mathcal{F}} \right. \\ \wedge \exists \langle x', y, t, m \rangle \in \mathcal{C} \exists \theta: \\ \left. (E_X^i \cup E_F^i \models t\theta) \wedge (X^i \cup F^i \models m\theta) \right. \\ \left. \wedge x' = (e', a') \wedge e'\theta = e \wedge a'\theta = a \right\} \quad (5)$$

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 conditions t and m hold, mimicking, thus, in the constitutive

level, the atomic nature of the environmental events [18]. Thus, the holding of m during many steps of the SAI history does not imply the holding of an event-status function assignment.

The rule 3 in the Fig. 3 defines the first-order constitution of an event-status function. If $\langle \text{put_tangible}(\text{launch_object}, 15, 20), \text{tom} \rangle \in E_X^2$ meaning that tom has put the a launch_object on the coordinates (15,20) of the table at the step 2, then the assignment $\langle \text{put_tangible}(\text{launch_object}, 15, 20), \text{evacuate}(\text{downtown}), \text{tom} \rangle$ holds in the step 2, i.e. $\langle \text{put_tangible}(\text{launch_object}, 15, 20), \text{evacuate}(\text{downtown}), \text{tom} \rangle \in f\text{-const}_e(\mathcal{F}, \mathcal{C}, X^0 \dots X^z, F^0 \dots F^z, 2)$.

3.4.3 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 status function s_1 counts as a status function s_2 , all the elements constituting s_1 constitute also s_2 . For example, if the constitutive rule `firefighter count-as security_expert` is added to the specification in Fig. 3, then the agent-status function `security_expert` is actually assigned to all the concrete agents carrying the status function `firefighter`.

Definition 11 (*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 $s\text{-const}_a$ below:

$$s\text{-const}_a(\mathcal{F}, \mathcal{C}, X^0 \dots X^z, F^0 \dots F^z, i) = \left\{ \langle x, y \rangle \mid \begin{aligned} &x \in A_X^i \wedge y \in \mathcal{A}_{\mathcal{F}} \\ &\wedge \exists \langle x', y, t, m \rangle \in \mathcal{C} \exists \theta \exists s \in \mathbb{N} \forall k \in [s, i]: \\ &(E_X^s \cup E_F^s \models t\theta) \wedge (X^k \cup F^k \models m\theta) \\ &\wedge x'\theta \in \mathcal{A}_{\mathcal{F}} \wedge \langle x, x'\theta \rangle \in A_F^i \end{aligned} \right\} \quad (6)$$

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 a_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 8)]. When the agent a_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. $\langle \text{bob}, \text{mayor} \rangle \in A_F^i$), then $\langle \text{bob}, \text{authority} \rangle \in s\text{-const}_a(\mathcal{F}, \mathcal{C}, X^0 \dots X^z, F^0 \dots F^z, i)$ (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 12 (*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 $s\text{-const}_s$ below:

$$\begin{aligned}
s\text{-const}_s(\mathcal{F}, \mathcal{C}, X^0 \dots X^z, F^0 \dots F^z, i) = & \left\{ \langle x, y \rangle \mid (x \in S_X^i \vee x = \varepsilon) \wedge y \in \mathcal{S}_{\mathcal{F}} \right. \\
& \wedge \exists \langle x', y, t, m \rangle \in \mathcal{C} \exists \theta \exists s \in \mathbb{N} \forall k \in [s, i]: \\
& (E_X^s \cup E_F^s \models t\theta) \wedge (X^k \cup F^k \models m\theta) \\
& \left. \wedge x'\theta \in \mathcal{S}_{\mathcal{F}} \wedge \langle x, x'\theta \rangle \in S_F^i \right\} \quad (7)
\end{aligned}$$

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 s_X , then this state carries also the status function $y \in \mathcal{S}_{\mathcal{F}}$ [subject to the conditions t and m , as in the first-order constitution (Definition 9)]. When s_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 $\langle \text{security_phase}(\text{downtown}, \text{emergency}), \text{insecure}(\text{downtown}) \rangle \in S_F^i$, then $\langle \text{security_phase}(\text{downtown}, \text{emergency}), \text{red_alert} \rangle \in s\text{-const}_s(\mathcal{F}, \mathcal{C}, X^0 \dots X^z, F^0, \dots, F^z, i)$ (for $i \leq z$).

Definition 13 (*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 $s\text{-const}_e$ below:

$$\begin{aligned}
s\text{-const}_e(\mathcal{F}, \mathcal{C}, X^0 \dots X^z, F^0 \dots F^z, i) = & \left\{ \langle e, y, a \rangle \mid e \in E_X^i \wedge y \in \mathcal{E}_{\mathcal{F}} \wedge a \in A_X^i \right. \\
& \wedge \exists \langle x', y, t, m \rangle \in \mathcal{C} \exists \theta: \\
& (E_X^i \cup E_F^i \models t\theta) \wedge (X^i \cup F^i \models m\theta) \\
& \left. \wedge x'\theta \in \mathcal{E}_{\mathcal{F}} \wedge \langle e, x'\theta, a \rangle \in E_F^i \right\} \quad (8)
\end{aligned}$$

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_X , then e_X carries also the status function $y \in \mathcal{E}_{\mathcal{F}}$ (subject to the conditions t and m , as in the first-order constitution (Definition 10)). The assignment of y to e_X holds while the assignment of $x'\theta$ to e_X holds.

Consider, for example, the event-status function *security_procedure* in the crisis scenario and 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, $\langle \text{put_tangible}(\text{launch_object}, 15, 20), \text{evacuate}(\text{downtown}), \text{tom} \rangle \in E_F^i$ and, as a consequence, by the introduced rule, $\langle \text{put_tangible}(\text{launch_object}, 15, 20), \text{security_procedure}, \text{tom} \rangle \in s\text{-const}_e(\mathcal{F}, \mathcal{C}, X^0 \dots X^z, F^0 \dots F^z, i)$ (for $i \leq z$) because (1) the term x of the rule is an event-status-function that (2) is already assigned to the event $\text{put_tangible}(\text{launch_object}, 15, 20)$.

3.5 Building the constitutive state

From the previous definitions, it is possible to define how the constitutive state of an institution is built. First, we introduce the function *const* below that, for an institution where the set of status function is \mathcal{F} and the set of constitutive rules is \mathcal{C} , returns all the SFAs that can be deduced given environmental and constitutive histories of size i :

$$\begin{aligned}
const(X^0 \dots X^i, F^0 \dots F^i) = & \langle f\text{-const}_a(\mathcal{F}, \mathcal{C}, X^0 \dots X^i, F^0 \dots F^i, i) \\
& \cup s\text{-const}_a(\mathcal{F}, \mathcal{C}, X^0 \dots X^i, F^0 \dots F^i, i), \\
& f\text{-const}_e(\mathcal{F}, \mathcal{C}, X^0 \dots X^i, F^0 \dots F^i, i) \\
& \cup s\text{-const}_e(\mathcal{F}, \mathcal{C}, X^0 \dots X^i, F^0 \dots F^i, i), \\
& f\text{-const}_s(\mathcal{F}, \mathcal{C}, X^0 \dots X^i, F^0 \dots F^i, i) \\
& \cup s\text{-const}_s(\mathcal{F}, \mathcal{C}, X^0 \dots X^i, F^0 \dots F^i, i) \quad (9)
\end{aligned}$$

Additions and revocations of SFAs may create conditions for new constitutions. In despite of that, every constitutive state F^i of a SAI history is *closed under constitution*, i.e. 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.

$$\begin{aligned}
F^*(X^0 \dots X^i, F^0 \dots F^i) = & \begin{aligned} & F^i \text{ if } F^i = F^{i'} \\ & F^*(X^0 \dots X^i, F^0 \dots F^{i'}) \text{ otherwise} \end{aligned} \\
\text{s.t. } F^{i'} = & const(X^0 \dots X^i, F^0 \dots F^i) \quad (10)
\end{aligned}$$

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 11. Notice that the environmental and normative states X and N do not change.

$$\frac{SAI_{Dyn}^i = \langle X^i, F^i, N^i \rangle \quad F^i \neq const(X^0 \dots X^i, F^0 \dots F^i)}{\langle X^i, F^i, N^i \rangle \rightarrow \langle X^i, F^*(X^0 \dots X^i, F^0 \dots F^i), N^i \rangle} \quad (11)$$

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.

3.6 Example of constitutive dynamics

To illustrate the constitutive dynamics, we consider the scenario introduced in Sect. 3.3.2 and the constitutive specification illustrated in Fig. 3. 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 summarised 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 (1) the downtown is on preventive phase of the crisis management and (2) the downtown has 200 inhabitants. By the constitutive rule 5, the institution considers the downtown as a secure zone. At this moment, the agent *bob* checks in the *table_mayor* and the agents *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).

Table 2 Evolution of environmental and constitutive states

Step	Environmental state (X)	Constitutive state (F)
1	$A_X = \{bob, tom, jim, ana\}$ $E_X = \{(checkin(table_maior), bob),$ $(checkin(table_fire_brigade), tom),$ $(checkin(table_fire_brigade), jim),$ $(checkin(table_fire_brigade), ana)\}$ $S_X = \{security_phase(downtown, preventive),$ $nb_inhabit(downtown, 200)\}$	$A_F = \{(bob, mayor), (tom, firefighter,),$ $(jim, firefighter), (ana, firefighter)\}$ $S_F = \{(security_phase(downtown, preventive),$ $secure(downtown))\}$
2	$A_X = \{bob, tom, jim, ana\}$ $E_X = \{(putTangible(launch_object, 15, 20), bob)\}$ $S_X = \{security_phase(downtown, preventive),$ $nb_inhabit(downtown, 200)\}$	$A_F = \{(bob, mayor), (tom, firefighter,),$ $(jim, firefighter), (ana, firefighter)\}$ $E_F = \{(putTangible(launch_object, 15, 20),$ $evacuate(downtown), bob)\}$ $S_F = \{(security_phase(downtown, preventive),$ $secure(downtown))\}$
3	$A_X = \{bob, tom, jim, ana\}$ $S_X = \{security_phase(downtown, emergency),$ $nb_inhabit(downtown, 50)\}$	$A_F = \{(bob, mayor), (tom, firefighter,),$ $(jim, firefighter), (ana, firefighter)\}$ $S_F = \{(security_phase(downtown, emergency),$ $insecure(downtown))\}$
4	$A_X = \{bob, tom, jim, ana\}$ $E_X = \{(putTangible(launch_object, 15, 20), tom),$ $(send_message(evacuation, downtown), jim)\}$ $S_X = \{security_phase(downtown, emergency),$ $nb_inhabit(downtown, 50)\}$	$A_F = \{(bob, mayor), (tom, firefighter,),$ $(jim, firefighter), (ana, firefighter)\}$ $E_F = \{(putTangible(launch_object, 15, 20),$ $evacuate(downtown), tom),$ $(send_message(evacuation, downtown),$ $(evacuate(downtown), jim))\}$ $S_F = \{(security_phase(downtown, emergency),$ $insecure(downtown))\}$
5	$A_X = \{bob, tom, jim, ana\}$ $S_X = \{security_phase(downtown, preventive),$ $nbInhabit(downtown, 50)\}$	$A_F = \{(bob, mayor), (tom, firefighter,),$ $(jim, firefighter), (ana, firefighter)\}$ $S_F = \{(security_phase(downtown, preventive),$ $secure(downtown))\}$

- **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).

4 Introducing norms in SAI

SAI assumes that the whole normative regulation is based on the constitutive state, for any normative model that is being used. The different normative models proposed in the literature—that might be part of SAI—are not explicitly conceived to be based on constituted elements. For each model, it is necessary to (1) semantically align the normative representations and the constitutive ones and (2) define how the dynamics of the social order that norms aim to achieve is animated by the dynamics of the constitutive state described in Sect. 3.4 [9]. For example, considering the norm “a bidder is obliged to bid”, it is necessary to define (1) how to monitor the norm taking into account all the agents considered as bidders, (2) how to proceed when obliged agents are no longer considered as bidders and (3) 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?).

The SAI constitutive state is already integrated with a normative model—namely the one proposed by Panagiotidi et al. [60]—as described in [25]. In this paper we present another integration—with Normative Programming Language (NPL) [48,49]. We choose NPL because it has a formal semantics as well as an implementation where we can test our approach. Since our goal is to evaluate the flexibility of our proposal, the focus of this section is to illustrate how an existing normative language can be integrated with SAI. For this reason, we take NPL as currently proposed and available, without introducing any change.¹⁰ In this integration, we first deal with the alignment between normative and constitutive representations and, then, describe how the normative dynamics is coupled with the constitutive one. Finally we illustrate how the normative regulation provided by NPL evolves based on the constitutive state.

4.1 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 first-order logic 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.¹¹ In this case, the activation condition represents an undesirable (failure) state that, when achieved, must be undone. 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.

¹⁰ An implementation of the NPL engine is available at <https://github.com/moise-lang/npl>.

¹¹ Regimentation is the strategy to lead individuals to behave as expected by preventing them to violate the norms [53,58].

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

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

Fig. 5 Examples of NPL norms

Fig. 6 Lifecycle of obligations in NPL [49]

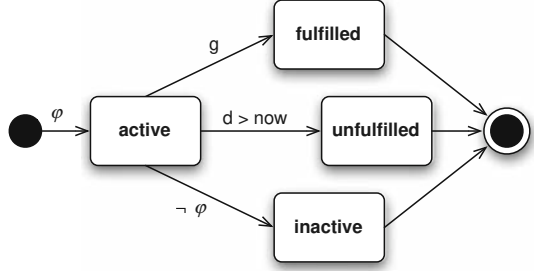


Figure 5 shows NPL norms using the status functions of Fig. 3 to specify that (1) the *mayor* is obliged to evacuate secure zones (*n1*) and (2) *firefighters* are obliged to evacuate insecure zones (*n2*).

An obligation is created, getting *active* when the activation condition φ is satisfied. An active obligation can become (1) *fulfilled*, when the goal g is achieved before the deadline d ; (2) *unfulfilled*, when the deadline d is satisfied before the fulfilment of g ; or (3) *inactive*, when the activation condition φ ceases to hold (Fig. 6).

The state of a NPL normative system is a tuple $\langle F, N, s, OS, t \rangle$ where (1) F is the set of facts considered in the evaluation of the norms, (2) N is a set of norms, (3) s is the state of the normative system, that can be sound, denoted by \top or failure (i.e. an undesirable state), denoted by \perp , (4) 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 $ost \in \{\text{active, fulfilled, unfulfilled, inactive}\}$, and (5) t is the current time considered by the normative management platform. Regarding to the obligations in OS , it is important to note that an obligation can be only in one state at a time, or, formally:

$$\langle o, ost \rangle \in OS \rightarrow \langle o, ost' \rangle \notin OS \quad (\text{s.t. } ost \neq ost') \quad (12)$$

4.2 Aligning normative and constitutive representations

Linking norms to the constitutive state requires to relate the semantics of the components of the norm to the semantics of the elements composing the constitutive state (and composing the normative state, when applicable).¹²

For a norm $n \in \mathcal{N}$, where $n = id: \varphi \rightarrow \psi$, we first explicitly define that activation condition of a norm is evaluated with respect to the whole constitutive state. Conditions over the whole constitutive state are expressed through sf-formulae $m_{\mathcal{F}} \in \mathcal{M}_{\mathcal{F}}$, that are a subclass of the m-formulae whose atoms are either event- and state-status functions or expressions of the type x is y .

¹² Assuming institutions composed of both constitutive and normative states, some elements of norms may be related to the very normative state. For example, a norm can be activated when other norm is violated [55].

Failures raised when $\psi = \text{fail}(r)$ are internally handled by the NPL engine and are not evaluated against the constitutive state. When the consequence is an $\text{obligation}(a, r, g, d)$, we explicitly define that $a \in \mathcal{A}_{\mathcal{F}}$, $g \in \mathcal{E}_{\mathcal{F}} \cup \mathcal{S}_{\mathcal{F}}$, and $r \in \mathcal{M}_{\mathcal{F}}$. The reasons for these decisions are:

- when coupled with SAI, norms refer to the status functions instead of to the environment. Thus, obligations are specified as directed to the agent-status function a instead of to concrete agents acting in the environment.
- the goal to be achieved is anchored to event- and state-status functions as, from the institutional perspective, all that the agents can do to behave as prescribed by the norms is to produce, in the environment, events and states that carry event- and state-status functions.
- the reason r , that determines the maintenance of an obligation in the *active* state, is also evaluated against the whole constitutive state.

The deadline d is expressed as time, that is internally managed by the NPL platform instead of being considered as part of the external “world” under regulation. Thus, the deadline d in the case of NPL is not related to the status functions and thus not considered in the integration with SAI.

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

4.3 Coupling normative and constitutive dynamics

In [49], norms are evaluated with respect to a set of facts. When the NPL dynamics is based on the SAI constitutive state, these facts are status function assignments composing the SAI constitutive state. As said before, norm activations can raise failures and obligations. Failures require regimentation, that concerns 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 Fig. 6 evolves based on the constitutive state.

Based on the original NPL operational semantics [49], the next sections describe 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 13–16 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 1 and 2. By the transition rule 13, 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 it 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 [49]. The notation $\stackrel{obl}{=}$ is used for equality of obligations ignoring the deadline in the comparison.

$$\frac{\text{norm id: } \varphi \rightarrow \psi \in N \quad \psi = o \quad F \models \varphi\theta \wedge a_X \text{ is } a \quad d\theta > t}{\frac{\neg \exists \langle o', \text{ost} \rangle \in OS : (o' \stackrel{obl}{=} o\theta \wedge \text{ost} \neq \text{active})}{\langle F, N, \top, OS, t \rangle \rightarrow \langle F, N, \top, OS \cup \langle o\theta, \text{active} \rangle, t \rangle}} \quad \text{s.t. } o = \text{obligation}(a, r, g, d) \text{ and } o\theta = \langle (a_X, a), r\theta \wedge a_X \text{ is } a, g\theta, d\theta \rangle \quad (13)$$

For example, considering the specification in Fig. 3, if the agents *bob* and *tom* carry the status function *firefighter* (i.e. $\{\langle \text{bob}, \text{firefighter} \rangle, \langle \text{tom}, \text{firefighter} \rangle\} \subseteq A_F$) and the *downtown* is in emergency phase of crisis, being thus insecure (i.e. $\langle \text{security_phase}(\text{downtown}, \text{emergency}), \text{insecure}(\text{downtown}) \rangle \in S_F$), then (1) $F \models \text{insecure}(\text{downtown})$, (2) $F \models \text{bob is firefighter}$, and (3) $F \models \text{tom is firefighter}$. Thus, considering the norm *n2* shown in Fig. 5, the following obligations are created:

$$\begin{aligned} &\langle \langle \text{bob}, \text{firefighter} \rangle, \text{insecure}(\text{downtown}) \wedge \text{bob is firefighter}, \text{evacuate}(\text{downtown}), d' \rangle \\ &\langle \langle \text{tom}, \text{firefighter} \rangle, \text{insecure}(\text{downtown}) \wedge \text{tom is firefighter}, \text{evacuate}(\text{downtown}), d' \rangle \end{aligned}$$

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* [49]. This is expressed by $F \not\models r'$ in the transition rule 14.

$$\frac{os \in OS \quad os = \langle o, \text{active} \rangle \quad F \not\models r'}{\langle F, N, \top, OS, t \rangle \rightarrow \langle F, N, \top, (OS \setminus \{os\}) \cup \{\langle o, \text{inactive} \rangle\}, t \rangle} \quad \text{s.t. } o = \langle (a_X, a), r', g', d' \rangle \quad (14)$$

The transition rule 14 implicitly captures the idea of obligations being directed to the concrete agents but being conditioned by the agent-status function assignments. Recall that, if $\langle (a_X, a), r', g', d' \rangle$ is an obligation produced by $\text{norm id: } \varphi \rightarrow \langle a, r, g, d \rangle$, then $r' = r\theta \wedge a_X \text{ is } a$ (cf. transition rule) 13. 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 $F \not\models a_X \text{ is } a$). For example, we can imagine that the agent *bob* is obliged to evacuate the downtown because it 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* loses this function.

Fulfilment Fulfilments are considered separately according to the nature of the goal to be achieved (event or state). The transition rules 15 and 16 deal respectively with fulfilments of active instances conditioned by events and by states.

$$\frac{os \in OS \quad os = \langle o, \text{active} \rangle \quad g' \in \mathcal{E}_{\mathcal{F}}}{\frac{\exists (e_X, a_X) \in E_X : F \models (e_X, a_X) \text{ is } g'}{\langle F, N, \top, OS, t \rangle \rightarrow \langle F, N, \top, (OS \setminus \{os\}) \cup \{\langle o, \text{fulfilled} \rangle\}, t \rangle}} \quad (15)$$

$$\frac{os \in OS \quad os = \langle o, \text{active} \rangle \quad g' \in \mathcal{S}_{\mathcal{F}} \quad F \models g'}{\langle F, N, \top, OS, t \rangle \rightarrow \langle F, N, \top, (OS \setminus \{os\}) \cup \{\langle o, \text{fulfilled} \rangle\}, t \rangle} \quad \text{s.t. } o = \langle (a_X, a'), r', g', d' \rangle \quad (16)$$

The transition rule 15 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 $F \models ((e_X, a_X)\mathbf{is}g')$, evaluated according to the Expression (2). By the transition rule 16, 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 expressions 1 and 2.

Unfulfilment 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 unfulfilment 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 17, defining how the normative state is moved from a consistent (\top) to a failure, undesirable (\perp) state, is the same as the original one introduced in [49].

$$\frac{n \in N \quad F \models \varphi \quad n_\psi = \text{fail}(_)}{\langle F, N, \top, OS, t \rangle \rightarrow \langle F, N, \perp, OS, t \rangle} \quad (17)$$

When failure normative states are produced, the NPL engine is conceived to roll 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. This issue is related to the very notion of regimentation as handled by the implementation of NPL: agents acting in the environment can produce undesirable states that are rolled back in the normative representation of the facts but that are not necessarily reverted in the environment.

The NPL strategy to deal with undesirable states mixes the notions of regulation and of institutional reality (even the later is not explicit in the NPL model). A norm where 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 $\text{fail}(r)$ 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 produce obligations to the agents to act in the environment producing environmental facts that bring the normative state consistent again.

4.4 Example of normative dynamics based on the constitutive one

Considering the proposed coupling, we illustrate the evolving of the normative regulation based on the constitutive dynamics illustrated in Sect. 3.6. In each step, the environmental state changes causing changes in the constitutive state (Table 2) and, as consequence, the normative state changes as summarised in the Table 3:

- **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.

Table 3 Evolution of the normative state

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

- **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.

5 Analysing stability, consistency, and flexibility in SAI

We have so far shown (1) how SAI builds the institutional reality (represented by the constitutive state) where the regulation is inserted and (2) how a specific normative model fits in this institutional reality to regulate MAS. This section analyses how basing norms on the constitutive state provides stability, consistency, and flexibility to the regulation in MAS.

5.1 Stability

SAI provides, through the status functions, the *institutional vocabulary* on top of which the regulation is specified. Norms do not specify nor manage the link between this vocabulary and the regulated environment. This is all done in the constitutive level. Norms are thus stable as they remain unchanged, referring to the status functions, even if the environment abstracted by the status functions changes.

Assuming that the environmental elements involved in the regulation are agents acting, events occurring, and states holding in the environment, the three kinds of status functions of SAI are enough to abstract the elements required by the norms. We use the example of Sect. 4.4, focusing on the norm 2 of Fig. 5, to illustrate stability with respect to these three kinds of elements:

- *Agents* The norm distributes obligations to all the agents carrying the status function of *firefighter* (Table 3—step 2). The norm does not need to specify every possible agent targeted by the obligation. The same norm would produce obligations to any additional agent carrying that status function.

- *Events* The norm defines obligations to the agents to evacuate zones. This single norm can be fulfilled through the production of two different events in the environment (Table 3—step 3). Additional events counting as evacuations could be added to the constitutive specification without any change in the norm.
- *States* The norm produces obligations when a zone is considered insecure. By the specification of Fig. 3, such condition is realised in the environment when a GIS defines that it is on emergency phase of crisis. If we add the constitutive rule “flooded (Zone) **count-as** insecure (Zone)”, meaning that a zone is insecure when GIS informs that it is flooded, the same norm 2, without any change, covers a new environmental state.

While the environmental elements are abstracted by the status functions, the way these abstracted elements are affected by the regulation is related to the way the norms are coupled with the constitutive state. The coupling proposed in Sect. 4 explicitly defines that (1) norms govern all the agents under the same constitution of agent-status function and (2) the activation, maintenance, and deactivation conditions, differently, point to a single constitution of status function. In the crisis example, a single norm targets all the agents counting as *firefighter* but its fulfilment requires that every firefighter produces (at least) one event counting as *evacuation*.

5.2 Consistency

The set of status functions of a constitutive specification provides a common vocabulary to be used to write the norms of an institution. More than providing the set of words, the institutional reality as conceived by SAI provides semantics to the institutional vocabulary through the typing of the status functions. This feature provides a consistent normative specification and dynamics, as discussed in the sequence.

5.2.1 Consistency in the normative specification

As the elements referred by the norms are typed as agent-, event-, or state status functions, it is possible to explicitly constrain the kind of element which the different components of the norms can refer to. Normative specifications become thus consistent. For example, it is possible to define that the “*a*” of NPL norms refers exclusively to agent-status functions while the element “*g*” refers to event- or state-status functions. Any reference of *a* to elements other than agent-status function and of *g* to elements other than event- or state-status functions makes the norm semantically wrong. For instance, if *firefighter* is an agent-status function and *evacuate* is an event-status function, then *evacuate* cannot be the bearer and *firefighter* cannot be the goal of a norm. Such consistency does not depend neither on the normative model nor on the expertise of the designer of the normative specification. Rather, it is provided by the proper link between the components of the norms and the different kinds of status functions (cf. Sect. 4.2).

5.2.2 Consistency in the normative dynamics

With typed status functions we can explicitly define relations between the different kinds of elements within the institutional reality and the different natures of their environmental counterparts. Thanks to these relations, the normative state is consistent with the environment. The constitution of the different kinds of status functions takes into account the particular dynamics of the different kinds of environmental elements (cf. functions in expressions 3–8).

Thus, norms being managed based on the constitutive state are consistent with the actual environment under regulation as, by referring to agent-, event-, and state-status functions, they range, respectively, over agents that are actually participating, events that are actually occurring, and states that are actually holding in the environment. Norms do not need to model nor monitor the environmental dynamics. All that is needed in the normative level is to define how the normative dynamics evolves based on the constitutive state, taking into account how the different components of the norms are related to the different kinds of status functions (cf. Sects. 4.2, 4.3).

5.3 Flexibility

Flexibility could be demonstrated by showing norms following different normative models coupled with the same constitutive state. This paper presents the coupling of the NPL model while a similar coupling of another normative model is presented in [25]. Norms following these models are applied to the same crisis scenario described in this paper, using the same constitutive specification of Fig. 3, considering the same environmental dynamics, and, as consequence, the same constitutive one, as the presented in Sect. 3.6. We have thus an application where norms following two different normative models are regulating the same scenario, based on the same institutional concepts.

Flexibility is achieved by the introduction of the constitutive level (as illustrated in Fig. 1), composed of status functions, between norms and environment. Norms, independent of the model they follow, are expressed in terms of status functions. Thus, all the norms within the institution share the same vocabulary. Furthermore, the semantics of the shared vocabulary, given in terms of status functions, is unified within the institution being also shared by all the norms taking part there. For example, all the norms in the institution, independent of the normative model, take *firefighter* as an agent-status function and *evacuate* as an event-status function. Finally, the grounding of the institutional vocabulary in the environment (i.e. the constitution of status functions) is managed by the introduced constitutive level, independent of the norms. For this reason, the constituted status functions provide a unified grounding in the environment of the elements referred by the norms.

As we can see, basing norms on the constitutive state makes the regulation (1) stable, as norms refer to *status functions*, that abstract the complexity and unpredictability of the environment, (2) consistent with the environment, as the *types* of the status functions allows us to coherently couple them with norms and also to coherently ground them in the environment and (3) flexible, as the status functions, as well as the constitutive state, provide a unified vocabulary, with unified semantics and unified grounding, accessible to all the norms regulating the system.

6 Limitations and future work

Spite of the discussed advantages, it is worth to remark some limitations of the SAI model. First of all, it is not possible to abstract within the institution elements of natures other than agents, events, and states. For example, while it is possible to define that placing an object on the coordinates (15,20) of the table counts as *evacuate(downtown)* (constitutive rule 3 in Fig. 3), it is not possible to define that those coordinates count as the *downtown*. Although this limitation does not prevent us to provide stability, consistency, and flexibility to norms that refer to agents, events, and states, some concepts referred by the norms, that are clearly related to the environment—as *downtown* in the example—remain unsituated. Dealing with

the institutional counterparts of elements other than agents, events, and states, is a future work.

The status functions of SAI are conceived to support the regulation provided by the norms. Besides norms, however, there are other social aspects in MAS, usually captured by metaphors inspired in human societies. There are, for example, organisations [35,51] and interaction protocols [68,79]. 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, [68] 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 *MOISE* model [51] 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. As these abstractions do not refer themselves to concrete elements in the environment, it seems suitable to use them in the context of institutions. Thus, a future work is to check whether and how the current state of SAI model can base the regulation provided by regulative representations other than norms, extending the model if necessary.

Some of those referred regulative abstractions capture collective social aspects. For example, groups of *MOISE* abstract sets of agents while missions abstract sets of states to be produced following a given sequence [51]. 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. The future work of inserting collective abstractions in institutions as conceived by SAI addresses another limitation of this work, as it covers only the abstraction of environmental elements at an individual level.

This paper considers a single institution gathering all the status function assignments that compose the constitutive state. It does not address the deployment of such an institution that, even centralised, could run in distributed nodes (e.g. a node running on the store system, a node running on the credit card system, a node running on the delivery system etc). Deploying artificial institutions requires to address, in the future, practical aspects, such as operational semantics, architecture, and implementation of interpreters for the proposed specification language.¹³ Furthermore, aspects such as the analysis of the cost of introducing a layer between norms and environment, as well as the scalability of the model, must be analysed. A further formal analysis based on the proposed formalism is also planned.

Theoretical aspects of the presented work were illustrated in this paper through examples from an application on crisis management. In these examples, SAI is proven to be useful to solve some issues [27–31]. As SAI is a general purpose model, applying it to other scenarios that can be modelled as open MAS, such as those pointed in Sect. 1, is a future work. Finally, as well as the coupling of SAI with the NPL (shown in this paper) and with the normative model proposed by [60] (shown in [25]), we plan to investigate the coupling with other normative models.

7 Conclusions

In this paper, we presented the SAI model as a way to provide stability, consistency, and flexibility to the normative regulation of MAS. We consider that norms are not the only element

¹³ Although a preliminary version of an interpreter is available at <http://sitartinst.sf.net> and was used to verify the feasibility of the proposal, several implementation concerns, as performance, are not verified yet.

necessary to regulate MAS. Rather, norms are conceived as part of artificial institutions that enable the institutional reality to which they refer.

The first contribution of this work is the conception of the abstractions to represent this institutional reality. This reality is conceived, in SAI, as an element that is neither part of the regulative abstractions nor of the environment. Rather, it is an element in itself within the institution. Just like norms are the building block of regulation, we introduce status functions as the building block of the institutional reality.

The relation between physical and social dimensions in MAS has been studied from many different perspectives, such as the institutional power emerging from the environmental circumstances [52], or contextual aspects of count as [4,5], or even the interoperability between implementations of physical and social dimensions [15,73]. In this direction, another contribution of this work is to define the use of count-as as a mean to *constitute status functions*. The notion of count-as in SAI is thus not just a link between environment and institution. Rather, it comprises the whole semantics related to the constitution of different kinds of elements within the institution, that is essential to build the SAI constitutive state.

This paper contributes also by conceiving the coupling of the NPL model with the status functions of SAI. Having regulation and constitution as independent processes within the institution requires to couple norms with the constitutive state. This clear separation provides flexibility as constitution is not tailored for a specific normative model. The required work of coupling norms with the constitutive state keeps the consistency as the particular semantics of the norms is aligned with the semantics of the status functions.

As said in the introduction, this paper focuses on open MAS, that are traditionally seen as those where the participating agents are not known in design time. Considering the openness feature, normative systems usually employ high level abstractions, such as *role*, to abstract of the concrete agents that are unknown when norms are designed [71]. But while openness is mainly addressed from this agent perspective, SAI enlarges the scope by proposing solutions to design and monitor norms ranging of an extensible, unknown in design time, set of environment elements. That is to say, from the normative perspective the system is open not only with respect to the participating agents but with respect to all the elements that can be abstracted under the notion of status function.

Acknowledgements This research supported by the Coordination for the Improvement of Higher Education Personnel (CAPES), linked to the Ministry of Education in Brazil (Grant PDSE 4926-14-5) and by the National Council for Scientific and Technological Development (CNPq), linked to the Ministry of Science and Technology in Brazil (Grants 448462/2014-1, 306301/2012-1).

References

1. Aldewereld, H., Álvarez Napagao, S., Dignum, F., & Vázquez-Salceda, J. (2010). Making norms concrete. In W. van der Hoek, G. A. Kaminka, Y. Lespérance, M. Luck, & S. Sen (Eds.), *Proceedings of the 9th international conference on autonomous agents and multiagent systems (AAMAS 2010)* (Vol. 1–3, pp. 807–814). Richland, SC: IFAAMAS.
2. Andrighetto, G., Governatori, G., Noriega, P., & van der Torre, L. W. N. (Eds.) (2013). Normative multi-agent systems. In *Dagstuhl follow-ups* (Vol. 4). Dagstuhl: Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik.
3. Aranda-Corral, G.A., Borrego-Díaz, J., & Martín, D. S. (2005). Iabastos: An intelligent marketplace for agricultural products. In Y. Demazeau (Eds.) et al. [34] (pp. 255–258). <https://doi.org/10.1007/978-3-319-18944-4>.
4. Artikis, A., Pitt, J., & Sergot, M. (2002). Animated specifications of computational societies. In *Proceedings of the first international joint conference on autonomous agents and multiagent systems (AAMAS 2002)* (pp. 1053–1062). New York, NY: ACM Press.

5. Artikis, A., Sergot, M. J., & Pitt, J. V. (2009). Specifying norm-governed computational societies. *ACM Transactions on Computational Logic (TOCL)*. <https://doi.org/10.1145/1459010.1459011>.
6. Balke, T., da Costa Pereira, C., Dignum, F., Lorini, E., Rotolo, A., Vasconcelos, W., & Villata, S. (2013). Norms in MAS: Definitions and related concepts. In G. Andrighetto (Eds.), et al. [2] (pp. 1–31). <https://doi.org/10.4230/DFU.Vol4.12111.1>.
7. Boella, G., Noriega, P., Pigozzi, G., & Verhagen, H. (Eds.). (2009). Normative multi-agent systems, 15.03.–20.03.2009. In *Dagstuhl seminar proceedings* (Vol. 09121). Dagstuhl: Schloss Dagstuhl - Leibniz-Zentrum für Informatik.
8. Boella, G., & van der Torre, L. (2004). An agent oriented ontology of social reality. In *Proceedings of formal ontologies in information systems (FOIS'2004)* (pp. 199–209). Amsterdam: IOS Press.
9. Boella, G., & van der Torre, L. (2006). Constitutive norms in the design of normative multiagent systems. In F. Toni, & P. Torroni (Eds.), *Computational logic in multi-agent systems, lecture notes in computer science* (Vol. 3900, pp. 303–319). Berlin: Springer. https://doi.org/10.1007/11750734_17.
10. Boella, G., & van der Torre, L. (2006). A logical architecture of a normative system. In L. Goble, & J. J. Meyer (Eds.), *Deontic logic and artificial normative systems, lecture notes in computer science* (Vol. 4048, pp. 24–35). Berlin: Springer. https://doi.org/10.1007/11786849_5.
11. Boella, G., van der Torre, L., & Verhagen, H. (2008). Introduction to the special issue on normative multiagent systems. *Autonomous Agents and Multi-Agent Systems*, 17(1), 1–10.
12. Boella, G., & van der Torre, L. W. N. (2004). Regulative and constitutive norms in normative multiagent systems. In D. Dubois, C. A. Welty, & M. A. Williams (Eds.), *Principles of knowledge representation and reasoning: Proceedings of the ninth international conference (KR2004)* (pp. 255–266). Whistler: AAAI Press.
13. Boella, G., van der Torre, L. W. N., & Verhagen, H. (Eds.). (2007). Normative multi-agent systems, 18.03.–23.03.2007. In *Dagstuhl seminar proceedings* (Vol. 07122). Dagstuhl: Internationales Begegnungs- und Forschungszentrum für Informatik (IBFI), Schloss Dagstuhl.
14. Brachman, R., & Levesque, H. (2004). *Knowledge representation and reasoning*. San Francisco, CA: Morgan Kaufmann Publishers Inc.
15. Bromuri, S., & Stathis, K. (2009). Distributed agent environments in the ambient event calculus. In A. S. Gokhale, & D. C. Schmidt (Eds.), *Proceedings of the third ACM international conference on distributed event-based systems, DEBS 2009, Nashville, TN*. ACM. <https://doi.org/10.1145/1619258.1619275>.
16. Campos, J., López-Sánchez, M., Rodríguez-Aguilar, J., & Esteva, M. (2009). Formalising situatedness and adaptation in electronic institutions. In J. Hübner, E. Matson, O. Boissier, & V. Dignum (Eds.), *Coordination, organizations, institutions and norms in agent systems IV, lecture notes in computer science* (Vol. 5428, pp. 126–139). Berlin: Springer.
17. Cardoso, H. L., & Oliveira, E. C. (2007). Institutional reality and norms: Specifying and monitoring agent organizations. *International Journal of Cooperative Information Systems*, 16(1), 67–95.
18. Cassandras, C. G., & Lafortune, S. (2006). *Introduction to discrete event systems*. Secaucus, NJ: Springer.
19. Castelfranchi, C. (2000). Engineering social order. In A. Omicini, R. Tolksdorf, & F. Zambonelli (Eds.), *Engineering societies in the agent world, first international workshop, ESAW 2000, Berlin, Germany, August 21, 2000, Revised Papers, lecture notes in computer science* (Vol. 1972, pp. 1–18). Berlin: Springer.
20. Cliffe, O., De Vos, M., & Padget, J. (2007). Answer set programming for representing and reasoning about virtual institutions. In K. Inoue, K. Satoh, & F. Toni (Eds.), *Computational logic in multi-agent systems, lecture notes in computer science* (Vol. 4371, pp. 60–79). Berlin: Springer. https://doi.org/10.1007/978-3-540-69619-3_4.
21. Criado, N., Argente, E., & Botti, V. J. (2011). Open issues for normative multi-agent systems. *AI Communications*, 24(3), 233–264. <https://doi.org/10.3233/AIC-2011-0502>.
22. Cuní, G., Esteva, M., García, P., Puertas, E., Sierra, C., & Solchaga, T. (2004). MASFIT: Multi-agent system for fish trading. In R. L. de Mántaras, & L. Saitta (Eds.), *Proceedings of the 16th European conference on artificial intelligence, ECAI'2004, including prestigious applicants of intelligent systems, PAIS 2004, August 22–27, 2004* (pp. 710–714). Valencia: IOS Press.
23. de Brito, M., Hübner, J. F., & Boissier, O. (2014). A conceptual model for situated artificial institutions. In N. Bulling, L. van der Torre, S. Villata, W. Jamroga, & W. Vasconcelos (Eds.), *Computational logic in multi-agent systems, lecture notes in computer science* (Vol. 8624, pp. 35–51). Cham: Springer. https://doi.org/10.1007/978-3-319-09764-0_3.
24. de Brito, M., Hübner, J. F., & Boissier, O. (2015). Bringing constitutive dynamics to situated artificial institutions. In F. C. Pereira, P. Machado, E. Costa, & A. Cardoso (Eds.), *Progress in artificial intelligence—17th Portuguese conference on artificial intelligence, EPIA 2015, Coimbra, Portugal. Proceedings, lecture notes in computer science* (Vol. 9273, pp. 624–637). Cham: Springer.
25. de Brito, M., Hübner, J. F., & Boissier, O. (2015). Coupling regulative and constitutive dimensions in situated artificial institutions. In M. Rovatsos, G. A. Vouros, & V. Julián (Eds.), *Multi-agent systems and*

- agreement technologies—13th European conference, EUMAS 2015, and third international conference, AT 2015, Athens, December 17–18, 2015, revised selected papers, lecture notes in computer science* (Vol. 9571, pp. 318–334). Cham: Springer.
26. de Brito, M., Hübner, J. F., & Bordini, R. H. (2013). Programming institutional facts in multi-agent systems. In H. Aldewereld, & J. S. A. Sichman (Eds.), *Coordination, organizations, institutions, and norms in agent systems VIII, lecture notes in computer science* (Vol. 7756, pp. 158–173). Berlin: Springer. https://doi.org/10.1007/978-3-642-37756-3_10.
 27. de Brito, M., Thévin, L., Garbay, C., Boissier, O., & Hübner, J. (2015). Institution artificielle située pour une aide à la régulation dans le cadre de la gestion de crises. In L. Vercouter & G. Picard (Eds.), *23es Journées Francophones sur les Systèmes multi-agents (JFSMA'15)* (pp. 133–142). Rennes: Cépaduès.
 28. de Brito, M., Thévin, L., Garbay, C., Boissier, O., & Hübner, J. (2016). Supporting flexible regulation of crisis management by means of situated artificial institutions. *Frontiers of Information Technology & Electronic Engineering*, 17(4), 309–324. <https://doi.org/10.1631/FITEE.1500369>.
 29. de Brito, M., Thévin, L., Garbay, C., Boissier, O., & Hübner, J. F. (2015). Situated artificial institution to support advanced regulation in the field of crisis management. In Y. Demazeau (Eds.), et al. [34] (pp. 66–79). https://doi.org/10.1007/978-3-319-18944-4_6.
 30. de Brito, M., Thévin, L., Garbay, C., Boissier, O., & Hübner, J. F. (2015). Situated regulation on a crisis management collaboration platform. In Y. Demazeau (Eds.), et al. [34] (pp. 267–270). <https://doi.org/10.1007/978-3-319-18944-4>.
 31. de Brito, M., Thévin, L., Garbay, C., Boissier, O., & Hübner, J. F. (2016). Institution artificielle située pour une aide à la régulation de la gestion de crises. *Revue d'Intelligence Artificielle*, 30(1–2), 185–209.
 32. Dastani, M., Grossi, D., Meyer, J. J. C., & Tinnemeier, N. (2009). Knowledge representation for agents and multi-agent systems, chap. In *Normative multi-agent programs and their logics* (pp. 16–31). Berlin: Springer.
 33. Dastani, M., van der Torre, L. W. N., & Yorke-Smith, N. (2012). 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* (pp. 17–34). Berlin: Springer. https://doi.org/10.1007/978-3-642-37756-3_2.
 34. Demazeau, Y., Decker, K. S., Pérez, J. B., & de la Prieta, F. (Eds.). (2015). *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, lecture notes in computer science* (Vol. 9086). Cham: Springer. <https://doi.org/10.1007/978-3-319-18944-4>.
 35. Dignum, V., Vázquez-Salceda, J., & Dignum, F. (2004). 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), August 19–23, 2004* (pp. 1498–1499). New York, NY: IEEE Computer Society. <https://doi.org/10.1109/AAMAS.2004.10250>.
 36. Esteva, M., Rodríguez-Aguilar, J. A., Sierra, C., Garcia, P., & Arcos, J. L. (2001). On the formal specifications of electronic institutions. In F. Dignum & C. Sierra (Eds.), *AgentLink, lecture notes in computer science* (Vol. 1991, pp. 126–147). Berlin: Springer.
 37. Farjam, M., & Kirchkamp, O. (2015). *Bubbles in hybrid markets—How expectations about algorithmic trading affect human trading*. CESifo Working Paper Series 5631, CESifo Group Munich.
 38. Fornara, N., Viganò, F., Verdicchio, M., & Colombetti, M. (2008). Artificial institutions: a model of institutional reality for open multiagent systems. *Artificial Intelligence and Law*, 16(1), 89–105.
 39. Franklin, S., & Graesser, A. (1997). Is it an agent, or just a program? A taxonomy for autonomous agents. In J. P. Müller, M. J. Wooldridge, & N. R. Jennings (Eds.), *Intelligent agents III agent theories, architectures, and languages: ECAI'96 Workshop (ATAL) Budapest, Hungary, August 12–13, 1996 proceedings, lecture notes in computer science* (Vol. 1193, pp. 21–35). Berlin: Springer. <https://doi.org/10.1007/BFb0013570>.
 40. Geiger, R. S. (2009). The social roles of bots and assisted editing programs. In D. Riehle, & A. Bruckman (Eds.), *Proceedings of the 2009 international symposium on Wikis, 2009, Orlando, Florida, USA, October 25–27, 2009*. New York, NY: ACM. <https://doi.org/10.1145/1641309.1641351>.
 41. Geiger, R. S. (2011). The lives of bots. In G. Lovink & N. Tkacz (Eds.), *Critical point of view: A Wikipedia Reader* (pp. 79–93). Amsterdam: Institute of Network Cultures.
 42. Grossi, D., Aldewereld, H., Vázquez-Salceda, J., & Dignum, F. (2006). Ontological aspects of the implementation of norms in agent-based electronic institutions. *Computational & Mathematical Organization Theory*, 12(2–3), 251–275.
 43. Grossi, D., & Dignum, F. (2004). From abstract to concrete norms in agent institutions. In *Formal approaches to agent-based systems, third international workshop, FAABS 2004, Greenbelt, MD, USA, April 26–27, 2004, Revised Selected Papers* (pp. 12–29). https://doi.org/10.1007/978-3-540-30960-4_2.
 44. Grossi, D., & Dignum, F. (2005). From abstract to concrete norms in agent institutions. In M. Hinchey, J. Rash, W. Truszkowski, & C. Rouff (Eds.), *Formal approaches to agent-based systems, lecture notes in*

- computer science (Vol. 3228, pp. 12–29). Berlin: Springer. https://doi.org/10.1007/978-3-540-30960-4_2.
45. Grossi, D., & Jones, A. J. I. (2014). Constitutive norms and counts-as conditionals. In D. Gabbay, J. Horty, X. Parent, R. van der Meyden, & L. Torre (Eds.), *Handbook of deontic logic and normative systems* (pp. 407–441). London, UK: College Publications.
 46. Helleboogh, A., Vizzari, G., Uhrmacher, A., & Michel, F. (2006). Modeling dynamic environments in multi-agent simulation. *Autonomous Agents and Multi-Agent Systems*, 14(1), 87–116. <https://doi.org/10.1007/s10458-006-0014-y>.
 47. Hindriks, F. (2012). But where is the university? *Dialectica*, 66(1), 93–113.
 48. Hübner, J. F., Boissier, O., & Bordini, R. H. (2010). A normative organisation programming language for organisation management infrastructures. In J. Padget, A. Artikis, W. Vasconcelos, K. Stathis, V. T. Silva, E. Matson, & A. Polleres (Eds.), *Coordination, organizations, institutions and norms in agent systems V, lecture notes in computer science* (Vol. 6069, pp. 114–129). Berlin: Springer. https://doi.org/10.1007/978-3-642-14962-7_8.
 49. Hübner, J. F., Boissier, O., & Bordini, R. H. (2011). A normative programming language for multi-agent organisations. *Annals of Mathematics and Artificial Intelligence*, 62(1–2), 27–53.
 50. Hübner, J. F., Boissier, O., Kitio, R., & Ricci, A. (2009). Instrumenting multi-agent organisations with organisational artifacts and agents. *Autonomous Agents and Multi-Agent Systems*, 20(3), 369–400. <https://doi.org/10.1007/s10458-009-9084-y>.
 51. Hübner, J. F., Sichman, J. S., & Boissier, O. (2007). Developing organised multiagent systems using the MOISE+ model: Programming issues at the system and agent levels. *International Journal of Agent-Oriented Software Engineering*, 1(3/4), 370–395.
 52. Jones, A., & Sergot, M. (1996). A formal characterisation of institutionalised power. *Logic Journal of IGPL*, 4(3), 427–443.
 53. Jones, A. J. I., & Sergot, M. (1993). Deontic logic in computer science. chap. In *On the characterization of law and computer systems: The normative systems perspective* (pp. 275–307). Chichester: Wiley.
 54. Kubicki, S., Lepreux, S., & Kolski, C. (2012). RFID-driven situation awareness on tangisense, a table interacting with tangible objects. *Personal and Ubiquitous Computing*, 16(8), 1079–1094. <https://doi.org/10.1007/s00779-011-0442-9>.
 55. López y López, F., & Luck, M., (2003). Modelling norms for autonomous agents. In *4th Mexican international conference on computer science (ENC 2003), September 8–12, 2003, Apizaco, Mexico* (pp. 238–245). Los Alamitos, CA: IEEE Computer Society Press.
 56. Merabet, G. H., Essaaidi, M., Talei, H., Abid, M. R., Khalil, N., Madkour, M., & Benhaddou, D. (2014). Applications of multi-agent systems in smart grids: A survey. In *2014 International conference on multimedia computing and systems (ICMCS)* (pp. 1088–1094). <https://doi.org/10.1109/ICMCS.2014.6911384>.
 57. Moses, Y., & Tennenholtz, M. (1995). Artificial social systems. *Computers and Artificial Intelligence*, 14(6), 533–562.
 58. Noriega, P., Chopra, A. K., Fornara, N., Cardoso, H. L., & Singh, M. P. (2013). Regulated MAS: Social perspective. In G. Andrighetto et al. [2] (pp. 93–133). <https://doi.org/10.4230/DFU.Vol4.12111.93>.
 59. Oren, N., Panagiotidi, S., Vzquez-Salceda, J., Modgil, S., Luck, M., & Miles, S. (2009). Towards a formalisation of electronic contracting environments. In J. F. Hübner, E. Matson, O. Boissier, & V. Dignum (Eds.), *Coordination, organizations, institutions and norms in agent systems IV, lecture notes in computer science* (Vol. 5428, pp. 156–171). Berlin: Springer.
 60. Panagiotidi, S., Álvarez-Napagao, S., & Vázquez-Salceda, J. (2013). 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* (pp. 346–363). Cham: Springer.
 61. Piunti, M., Boissier, O., Hübner, J. F., & Ricci, A. (2010). 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), CEUR workshop proceedings* (Vol. 627). CEUR-WS.org.
 62. Ricci, A., Piunti, M., & Virolì, M. (2011). Environment programming in multi-agent systems: An artifact-based perspective. *Autonomous Agents and Multi-Agent Systems*, 23(2), 158–192. <https://doi.org/10.1007/s10458-010-9140-7>.
 63. Ruiters, D. W. (1997). A basic classification of legal institutions. *Ratio Juris*, 10(4), 357–371.
 64. Russell, S., & Norvig, P. (2003). *Artificial intelligence: A modern approach* (2nd ed.). Englewood Cliffs, NJ: Prentice-Hall.
 65. Searle, J. (1995). *The construction of social reality*. New York: Free Press.

66. Searle, J. (2009). *Making the social world: The structure of human civilization*. Oxford: Oxford University Press.
67. Shoham, Y., & Tennenholtz, M. (1992). On the synthesis of useful social laws for artificial agent societies (preliminary report). In *Proceedings of the 10th national conference on artificial intelligence. San Jose, CA, July 12–16, 1992* (pp. 276–281).
68. Singh, M. P. (2011). 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* (Vols. 1–3, pp. 491–498). Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems.
69. Smith, B., & Searle, J. (2003). The construction of social reality: An exchange. *American Journal of Economics and Sociology*, 62(1), 285–309.
70. Thévin, L., Badeig, F., Dugdale, J., Boissier, O., & Garbay, C. (2015). Un système multi-agent normatif hybride pour l'interaction mixte. *Revue d'Intelligence Artificielle*, 29(3–4), 453–482. <https://doi.org/10.3166/ria.29.453-482>.
71. Tinnemeier, N., Dastani, M., & Meyer, J. J. (2009). Roles and norms for programming agent organizations. In *Proceedings of the 8th international conference on autonomous agents and multiagent systems—AAMAS '09* (Vol. 1, pp. 121–128). Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems. <http://dl.acm.org/citation.cfm?id=1558013.1558029>.
72. Tinnemeier, N. A. M., Dastani, M. M., Meyer, J. J. C., & van der Torre, L. (2009). Programming normative artifacts with declarative obligations and prohibitions. In *Web intelligence and intelligent agent technology, IEEE/WIC/ACM international conference on* (Vol. 2, pp. 145–152). <https://doi.org/10.1109/WI-IAT.2009.144>.
73. Urovi, V., Bromuri, S., Stathis, K., & Artikis, A. (2010). Towards runtime support for norm-governed multi-agent systems. In *Principles of knowledge representation and reasoning: Proceedings of the twelfth international conference, KR 2010, Toronto, Ontario, Canada, May 9–13, 2010*.
74. Viganò, F., & Colombetti, M. (2007). Specification and verification of institutions through status functions. In P. Noriega, J. Vquez-Salceda, G. Boella, O. Boissier, V. Dignum, N. Fornara, & E. Matson (Eds.), *Coordination, organizations, institutions, and norms in agent systems II, lecture notes in computer science* (Vol. 4386, pp. 115–129). Berlin: Springer. https://doi.org/10.1007/978-3-540-74459-7_8.
75. Viganò, F., & Colombetti, M. (2008). Model checking norms and sanctions in institutions. In J. S. A. Sichman, J. Padget, S. Ossowski, & P. Noriega (Eds.), *Coordination, organizations, institutions, and norms in agent systems III, lecture notes in computer science* (Vol. 4870, pp. 316–329). Berlin: Springer. https://doi.org/10.1007/978-3-540-79003-7_23.
76. Vos, M. D., Balke, T., & Satoh, K. (2013). 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* (pp. 1157–1158). Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems.
77. Weyns, D., Omicini, A., & Odell, J. (2007). Environment as a first-class abstraction in multiagent systems. *Autonomous Agents and Multi-Agent Systems*, 14(1), 5–30. <https://doi.org/10.1007/s10458-006-0012-0>.
78. Wooldridge, M., & Jennings, N. (1995). Agent theories, architectures, and languages: A survey. In M. Wooldridge, & N. Jennings (Eds.), *Intelligent agents: ECAI-94 workshop on agent theories, architectures, and languages Amsterdam, The Netherlands August 8–9, 1994 proceedings, lecture notes in computer science* (Vol. 890, pp. 1–39). Berlin: Springer. https://doi.org/10.1007/3-540-58855-8_1.
79. Zatelli, M. R., & Hübner, J. F. (2012). A unified interaction model with agent, organisation, and environment. In *Anais do IX Encontro Nacional de Inteligencia Artificial (ENIA@BRACIS)*. Curitiba.