A Metamodel for Verifying Institutions

Francesco Viganò

Università della Svizzera italiana, via G. Buffi 13, 6900 Lugano, Switzerland
francesco.vigano@lu.unisi.ch

Abstract

To investigate the interdependencies existing among deontic positions (like powers and obligations) and the ontology defined by an institution, we have proposed to *model* institutions in terms of status functions imposed on agents and defined as aggregates of deontic positions. In this paper we present a metamodel of institutional reality which introduces a set of concepts necessary to describe an institution and their intended meaning. A main advantage of our approach resides in the fact that institutions modelled in terms of such concepts can be *verified* by applying model checking techniques. In particular, in our framework it is possible to state and verify a set of properties stemming from our metamodel to enhance the development of sound institutions.

1 Introduction

Institutions and normative systems have been put forward as a means for regulating open interaction systems where agents' internal states cannot be accessed or agents are implemented by different parties. In such systems, *norms* play a fundamental role because they create positive expectations in the outcomes of interactions and make more predictable the behavior of other agents which are assumed to be autonomous [7, 15, 9, 10, 22, 26]. But while in [7] institutions define only norms, following [25] in [9] we have suggested that institutions also describe the *ontology* of the interaction context. For instance, the institution of the English Auction defines the very concept of winning an auction, which also implies that the winner ought to follow a set of norms.

To investigate the interdependencies existing among deontic positions (like institutionalized powers [16] and obligations) and the ontology defined by an institution, in [28] we have proposed to *model* institutions in terms of status functions imposed on agents and defined as aggregates of deontic positions. Our main tenet is that institutional facts are such only because they provide agents with possibilities of actions which are attributed to them only thanks to the common agreement of a community of agents. In particular, we characterize institutional events in terms of what status functions they impose or revoke, which reflect the powers and obligations created or cancelled in the system.

Once we have formalized an institution, we have also to ensure that it is sound and allows agents to reach the desired states of affairs. Furthermore, as soon as institutions become complex, without the aid of an automated techniques it is prohibitive to foresee all possible evolutions and states in which a system may evolve. For this reason in [28] we have defined FIEVeL (*F*unctions for *I*nstitutionalized *E*nvironments *Ve*rification *L*anguage), a high level language to model institutions in terms of status functions and which is amenable to model checking [4], either by translating it into the input language of an existing tool (e.g. Promela, the input language of SPIN [14] as in [28]) or by implementing a new model checker.

In this paper we focus our attention on the definition of a metamodel of institutions which defines a set of legal and philosophical concepts that we perceive as essential to describe institutional reality. In our framework, institutions are described with FIEVeL, a modelling language which provides a concrete syntax to formalize institutions in terms of such concepts. For this reason we say that such set of concepts and their relations define a metamodel [17], since they constitute a model of our modelling language. On the other hand, the institutional metamodel represents the *upper ontology* of institutional reality, since it introduces concepts that are extended to describe the ontology of institutions. For instance, any domain-dependent status function extends the notion of status function, which is abstractly defined as an aggregate of deontic position, by detailing what powers and obligations are associated to it.

The introduction of a metamodel allows us to define a library of domain-independent properties which not necessarily affect the functionality of an institution, but reflect the intended meaning of institutional concepts. For instance, if we empower agents participating to an auction to make bids but we always forbid them to do so, we obtain that it may be the case that agents make offers, but they will always violate the system of norms. Although such institution is functional (it allows agents to make bids), it is clearly irrational.

The remainder of this paper is structured as follows. Section 2 presents our metamodel, Section 3 discusses few domain-independent properties which can be verified by the tool presented in Section 4. In Section 5 we provide an example of verification activities that can be carried out in our framework by verifying the *institution of property* as it has been modelled and analyzed in [5], and finally in Section 6 we provide a comparison of our approach with related works.

2 The Institutional Metamodel

Many researches on institutions and normative systems [5, 9, 10, 22, 26] share several common or strongly related notions such as the concepts of role, norms, and institutionalized powers [16]. In this section we introduce our metamodel of institutions, that is, the set of concepts that we perceive as essential to specify an institution, the relationships existing between them, and their intended meaning. For the sake of brevity, we focus our attention on those aspects that are more relevant for the definition of what we call *domain-independent* properties, which, as we will see in Section 3, reflect important aspects of the institutional metamodel.

We express the semantics of the metamodel in terms of an order many-sorted first-order logic with temporal operators (OMSFOL). An order many-sorted first-order temporal logic is defined on a tuple $\mathfrak{S} = \langle \mathbf{\Sigma}, \leq_\Sigma, \mathbf{V}, \mathbf{C}, \mathbf{F}, \mathbf{P}, \xi \rangle$, which constitutes the *signature* of the logic, where $\mathbf{\Sigma}$ is a finite nonempty set of *sort symbols*; $\leq_\Sigma$ is a partial order on $\mathbf{\Sigma}$ determining a *hierarchy of sorts*; $\mathbf{V}$ is a finite set of (individual) *variables*, including a denumerable variables for every sort; $\mathbf{C}$ is a finite set of (individual) *constants*; $\mathbf{F}$ is an finite set of *function symbols*; $\mathbf{P}$ is an finite set of *predicate symbols*; ξ is a function that assigns a sort to every variable and every constant, and a signature (i.e. a sequence of sorts) to every function symbol and every predicate symbol; signatures of predicate symbols may be empty sequences, while signatures of function symbols have at least one component. Sets $\mathbf{\Sigma}$, $\mathbf{V}$, $\mathbf{C}$, $\mathbf{F}$, and $\mathbf{P}$ are mutually disjoint and we will write $\sigma_1 \leq_\Sigma \sigma_2$ to say that σ_1 is a *subsort* of σ_2.

Given sorts $\mathbf{\Sigma}$, the set $\mathbf{T}_\sigma$ of *terms of sorts* σ is the smallest set such that:

- $v \in \mathbf{T}_\sigma$ if $v \in \mathbf{V}$ and $\xi(v) \leq_\Sigma \sigma$
- $c \in \mathbf{T}_\sigma$ if $c \in \mathbf{C}$ and $\xi(c) \leq_\Sigma \sigma$
- $f(t_1, ..., t_n) \in \mathbf{T}_\sigma$ if $f \in \mathbf{F}$, $\xi(t_i) \leq_\Sigma [\xi(f)]_i$ for $1 \leq i \leq n$ and $[\xi(f)]_0 \leq_\Sigma \sigma$

where $[\xi(q)]_i$ refers to the i-th sort of the signature of a predicate or function symbol q.

The set $\mathbf{T}$ of *terms* is the union of the sets $\mathbf{T}_\sigma$ for all $\sigma \in \mathbf{\Sigma}$ and the set $\mathbf{A}$ of *atomic formulae* is the smallest set such that:

- $(t_1 = t_2) \in \mathbf{A}$ if there exists sort σ such that $\xi(t_1) \leq_\Sigma \sigma$ and $\xi(t_2) \leq_\Sigma \sigma$;
- $P(t_1, ..., t_n) \in \mathbf{A}$ if $P \in \mathbf{P}$ and $\xi(t_i) \leq_\Sigma [\xi(P)]_i$ for $1 \leq i \leq n$

The set of *formulae* is defined according to the following grammar:

$$\varphi ::= \alpha \mid \neg\varphi \mid \varphi \wedge \varphi \mid \mathbf{X}\varphi \mid [\varphi \mathbf{U} \varphi] \mid \mathbf{E}\varphi$$

where α is an atomic formula. Expressions *true*, *false*, $(\varphi \vee \psi)$, $(\psi \rightarrow \varphi)$, $(\varphi \leftrightarrow \psi)$, $\exists x\varphi$, $\exists_{\leq n} x\varphi$, $\exists_{\geq n} x\varphi$, $\mathbf{F}\varphi$, $\mathbf{G}\varphi$, $\mathbf{A}\varphi$, and $t_1 = t_2$ are introduced as abbreviations as usual.

The semantics of an order many-sorted first-order temporal logic is defined as usual [20] by providing a set of states, a *total* transition relation among states, a set of domains (one for each sort), and an interpretation function I which maps, for each state, constants to individuals, and function and predicate symbols to functions and relations on domains.

Despite OMSFOL models and formulae can be translated into classical first-order logic with temporal operators (FOL) or, under certain conditions, into temporal propositional logic like CTL* [4] (see Section

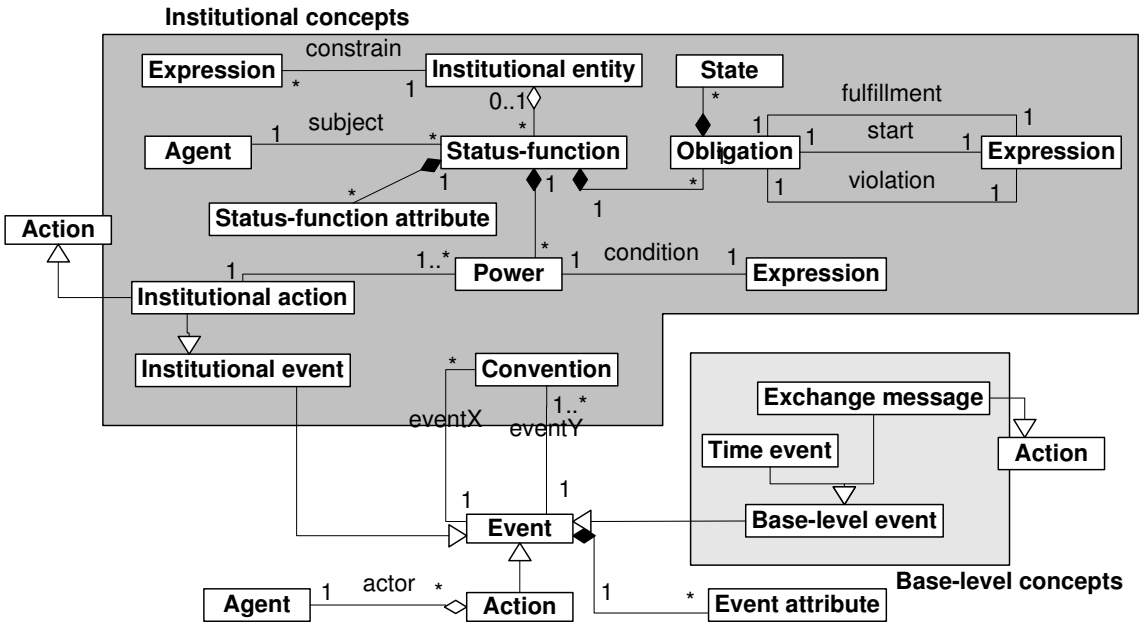

Figure 1: The institutional metamodel.

4), we adopt OMSFOL for two main reasons: i) it represents an abbreviated form for long and complex FOL or CTL* formulae and ii) OMSFOL guarantees syntactic type checking of formulae. Moreover, institutions describe rules that typically are independent of the number of agents, objects, etc. involved in the interaction, which can be naturally expressed by allowing quantification over sorts.

In the remainder of this section we will introduce the semantics of the institutional metamodel by explaining what sorts, functions, and predicates are induced by each institutional concept. For predicates and functions we also provide their signature ξ. Figure 1 depicts some of the main sorts used in our approach (e.g. status function, obligation, event) and their relations, which are typically represented by introducing predicates or functions. For instance, relation *actor* is reflected in our logic by function $actor$, which refers to the actor of the current action (see below). Finally, we report a set of axioms which characterize institutional reality by imposing a set of restrictions A on the admissible valuations of institutional models [1]. In the sequel we will represent such restrictions in terms of OMSFOL formulae (corresponding to LTL formulae), while in Section 3 we will introduce a set of properties that can be translated into CTL formulae.

Before starting the analysis of the concepts which constitute our metamodel, it is worth to mention few basic sorts like *integers*, *agents* (σ_{AID}), and *objects*, which are introduced to describe types of domain-dependent attributes associated to status functions and events. Designers can also define context dependent basic sorts.

As shown in Figure 1, our metamodel is based on the notion of agent status function, that is, a status imposed on an agent and recognized as existing by a set of agents. Typical examples of status functions are not only the concept of auctioneer, participant, or winner of an auction, but also being the owner of a good, being the husband or the wife of somebody. The concept of status function shares several features with the concept of role as it has been discussed in the literature (refer to [2] for an overview). Despite that, we prefer to use the term "status function" for three reasons: (i) the term role has been used with different meanings and it has been characterized in terms of very different concepts such as mental states, tasks, duties, etc; (ii) the term status function better represents the fact that we are concerned with status whose existence depends on those agents that recognize them as existing and which are assigned to agents to create new institutional powers or to regulate their use; (iii) the concept of status function is broader than the concept of role as used, for example, in [2]. In fact, it seems to be difficult to describe in terms of a "preexisting organization" being the owner of a good or being under age, while it is quite natural to regard them as status functions imposed by a group of agents.

Status functions induce sort σ_{SF}, which represents the supertype of all status functions described by an institution. σ_{SF} defines function $subject$ which refers to the agent the status function has been assigned to ($\xi(subject) = \langle \sigma_{AID}, \sigma_{SF} \rangle$) and predicate $assigned$ ($\xi(assigned) = \langle \sigma_{SF} \rangle$) which evaluates to true

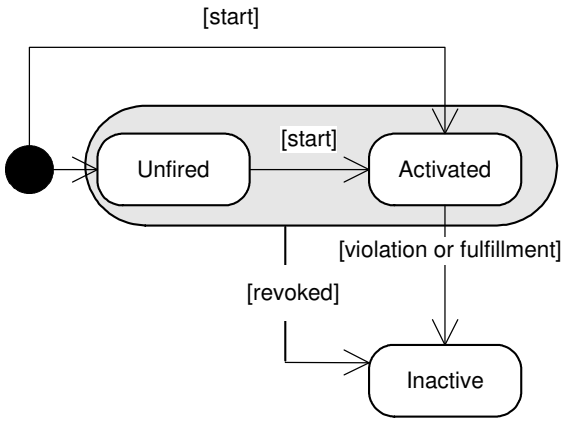

Figure 2: Life cycle of an obligation.

when a status function is currently assigned. Status functions are imposed (or revoked) when an institutional event happens (see below), otherwise they continue to be assigned (unassigned).

Status functions are possibly empty aggregates of deontic positions that can be expressed in terms of two main concepts, *institutionalized power* [16] [1] and *obligations*. Obligations induce sort σ_o whose individuals reify norms of institutions in such a way that it is possible to classify institutional states with respect to agents' compliance with norms, or adopting the terminology used in [19, 23, 26], it is possible to classify states as *red* (violating at least a norm) or *green states* (all norms are respected). Obligations can be also used to express prohibitions by specifying suitable violation expression, while we do not define a specific construct to explicitly represent the fact that an agent is permitted to perform an action as in [5, 10, 22]. Instead, we consider that every action, if it is not prohibited, is also permitted.

Given sort σ_{state}, which introduces constants $unfired$, $activated$, and $inactive$, obligation sort σ_o is characterized by function $state$ ($\xi(state) = \langle \sigma_{state}, \sigma_o \rangle$) and by a set of predicate (*start*, *fulfillment*, and *violation* of signature $\xi(violation) = \langle \sigma_o \rangle$) which are used to specify conditional obligations and when a norm should be considered fulfilled or violated according to the state machine represented in Figure 2. An obligation is created because a status function is imposed, changes its state when its conditions are satisfied, and eventually reaches a final state (*inactive*) either because it is fulfilled (violated) or because it is associated to a revoked status function. Figure 2 graphically describes a set of axioms which regulate the temporal evolution of an obligation individual: for instance, the following axiom states that if an obligation has been $activated$ and in the following state $violation$ or $fulfillment$ are evaluated to true, then the next state is $inactive$:

$$\mathbf{G}\forall o \forall sf(state(o) = activated \wedge \mathbf{X}(assigned(sf) \wedge ofStatusFunction(o) = sf \\ \wedge (violation(o) \vee fulfillment(o))) \rightarrow \mathbf{X}(state(o) = inactive)) \quad (1)$$

To record whether an obligation has been violated, we introduce predicate $violated$ of signature $\xi(violated) = \langle \sigma_o \rangle$. Predicate $violated$ is evaluated to $true$ in a given state if an obligation was $activated$ and the $violation$ expression is evaluated to $true$:

$$\mathbf{G}\forall ob(state(ob) = activated \wedge \mathbf{X}violation(ob) \rightarrow \mathbf{X}violated(ob)) \quad (2)$$

Given that at the moment our metamodel does not provide any support for the definition of recovery policies or sanctions [19], we assume that once an obligation has been violated, predicate $violated$ is always evaluated to true. Formally:

$$\mathbf{G}\forall ob(violated(ob) \rightarrow \mathbf{X}violated(ob)) \quad (3)$$

[1] In [9, 28] we used to name institutionalized power "authorization", but to enhance a comparison with other works we adopt the more widely used term "power" [16, 5, 22, 26].

The definition of axioms that regulate the temporal evolution of an obligation and when a norm is violated allows us to automatically classify states as *red* or *green* with respect to every norm, while in [23] the designer is required to manually specify which states are *red* or *green*.

According to Figure 1, obligations are associated to status functions and not directly to agents. In general, an agent is responsible for the state of affairs described by a fulfillment or violation expression because it has been assigned a specific status function. For this reason we introduce function $liable(ob)$, which is defined as follows:

$$liable(ob) \leftrightarrow subject(ofStatusFunction(ob))$$

where given an obligation, function $ofStatusFunction$ ($\xi(ofStatusFunction) = \langle \sigma_{SF}, \sigma_o \rangle$) returns the associated status function.

The metamodel defines two kinds of events, *base-level events* (σ_{BE}) and *institutional events* (σ_{ie}), which are characterized by an ontological difference: while the former exist because they reflect changes that are produced in the physical world or that are relative to lower level institutions, like *time* events and *exchange-message* events, the latter exist because they are recognized as existing by a community of agents and cannot be directly produced by the environment or by an agent [25, 9]. Despite that, we define both types of events as subsort of sort *event* (σ_{ev}) which defines predicate $happens$ ($\xi(happens) = \langle \sigma_{ev} \rangle$) and which is evaluated to true when event ev has caused the last transition. Analogously, sort σ_{ia} represents common features of base-level actions and of institutional actions, e.g. "actions are always performed by an $actor$" ($\xi(actor) = \langle AID, \sigma_{act} \rangle$). For convenience we define predicate $done$ as an abbreviation to say that "agent x has performed action act":

$$done(act, x) \leftrightarrow (happens(act) \wedge actor(act) = x)$$

It is important to observe that in the literature only agent actions have been considered relevant to describe institutions [5, 9, 22, 26], and the attention has been focused on a single action type, namely the act of exchanging a message [7, 9]. In contrast, we are also interested in modelling the institutional effects of events that are not generated by agents, like for instance time events, which not only are important for the management of obligations, but also may count as institutional events. For instance, in most cultures the 18th birthday imposes new status functions on a person.

We regard time aspects in two distinct ways: (i) as in classical temporal logic, to define *qualitative* aspects (it is always the case that a revoked status function is associated only to inactive obligations), and (ii) as in RTTL [21] to express *quantitative* aspects (e.g. an obligation will be violated in 2 time instants since now). To define quantitative aspects, we consider the perspective of a single centralized component which makes the institutional (normative) state evolve: therefore, we consider the existence of a unique timer that generates time events such that two consecutive time events t_i and t_{i+1} may be separated by a sequence (possible empty) of other base-level events, which are assumed to occur at time t_i. Hence the institutional state may change due to the occurrence of other events even if the timer has not generated new ticks of the clock. Although multiagent systems are distributed in nature, which makes problematic the assumption of unique notion of time or the adoption of an architecture where a single centralized manager control the evolution of the institutional state, to cope with the state explosion problem [4] we decide to introduce such simplification. It is worth noticing that centralized managers of the institutional states have been also proposed in several prototypes of institutions [6, 11] and normative systems [8].

Institutional events are defined by an institution and represent changes in the institutional environment. Essentially, their occurrence means that new status functions have been imposed on agents or existing status functions have been revoked. Sort σ_{ie} defines two predicates, $prec$ and eff which express the preconditions and effects of an institutional event. More precisely, for every institutional event $prec$ ($\xi(prec) = \langle \sigma_{ie} \rangle$) describes a condition that must be satisfied before institutional event ie occurs, while eff describes a condition satisfied after ie has occurred. But how agents recognize when an institutional event happens?

Following [25], we regard the occurrence of an institutional event by subordinating it the occurrence of another event conventionally associated to it. We represent the existence of a convention[2] among two events by introducing predicate $conv$ which associates the occurrence of an event to the occurrence of an institutional event ($\xi(conv) = \langle \sigma_{ev}, \sigma_{ie} \rangle$). We are now in position to define under what conditions

[2] As studied in [16, 12] conventions are related to the counts-as relation, which tipically is defined relative to a certain institution. The extension of our approach to the treatment of multiple institutions is an interesting topic of research that we intend to tackle in our future works. For the purpose of this paper we limit our attention to a single institution, and therefore we omit the institution in which such relation holds.

an institutional event ie which is not also an institutional action may happen. An institutional event ie conventionally bounded to event ev happens when:

$$\mathbf{G}\forall ie \exists ev(prec(ie) \wedge conv(ev, ie) \wedge \mathbf{X}happens(ev) \rightarrow \mathbf{X}happens(ie)) \tag{4}$$

Instead, in the case of institutional actions a further condition must be satisfied, namely, the actor must be empowered to perform the institutional action [16]. According to Figure 1 status functions are constituted by a set of powers, which are represented through predicate $empowered$ ($\xi(empowered) = \langle \sigma_{SF}, \sigma_{ia} \rangle$) which means that status function sf is empowered to perform institutional actions of type ia. An institutional action happens when:

$$\begin{aligned}\mathbf{G}\forall ia \forall aid \exists act(prec(ia) \wedge \exists sf(subject(sf) = aid \wedge empowered(sf, ia) \wedge assigned(sf)) \\ \wedge \mathbf{X}(happens(act) \wedge actor(act) = aid \wedge conv(act, ia)) \rightarrow \mathbf{X}(happens(ia))\end{aligned} \tag{5}$$

Assuming that institutions are asynchronous systems and according to axiom (5), which ensures that if the agent that brings about action act is empowered to perform ia, then action ia is successfully performed, we require that given two action symbols act_1 and act_2, if they are conventionally bounded and if they happen, the actor must be the same. Formally:

$$\begin{aligned}\mathbf{G}\forall act_1 \forall act_2(conv(act_1, act_2) \wedge happens(act_1) \wedge happens(act_2) \rightarrow \\ actor(act_1) = actor(act_2))\end{aligned} \tag{6}$$

Our treatment of the conventional generation of events extends the treatment presented in [9], since we do not take as our unique primitive to specify conventions the act of exchanging a message. Instead, any event can be used to define a new convention. In particular, our metamodel allows institutional events to be conventionally associated to other institutional events. For instance, the institutional act of transfer the possession may be conventionally bounded to the act of transfer the property.

3 Domain-Independent Properties

Once an institution has been defined in terms of the concepts described by our metamodel and the corresponding model has been generated by a model checker, there are two kinds of properties that a designer may want to verify, *domain-independent properties* and *domain-dependent properties* [28]. Domain-dependent properties stem from peculiar features of the specified model and regard the *functionality* of an institution: for instance, we may want to ensure that an auctioneer cannot win an auction. Instead, domain-independent properties are defined to guarantee the *rationality* of an institution with respect to the intended semantics of the concepts provided by our metamodel. Notice that, in contrast with axioms discussed in Section2 which characterize the semantics of institutional concepts and cannot be falsified, domain-independent properties are verified by a model checker and may be *unsatified* by institutions. In this case we say that an institution is irrational with respect to such properties. For instance, a rational institution should be characterized by the fact that every institutional event must eventually happen in at least an execution:

$$\forall ie \ \mathbf{EF}happens(ie) \tag{1}$$

where ie is a variable of type σ_{ie}. If this property does not hold, then it means that either the preconditions of an institutional event are never met or that the designer has not defined the necessary powers or conventions for its performance.

Analogously, given a convention which relates events of type x and y, it should be the case that there exists a path where eventually both of them contemporary happen:

$$\forall ev_x \forall ev_y(conv(ev_x, ev_y) \rightarrow \mathbf{EF}(happens(ev_x) \wedge happens(ev_y))) \tag{2}$$

If this property does not hold, it means that agents cannot exploit the convention binding events of type x to events of type y, for instance because preconditions of such events are never contemporary satisfied or, in the case of actions, because status functions empowered to execute them are never assigned to an agent.

Let us now move our attention to a set of properties that should characterize norms. Norms are introduced in open multiagent systems to constrain possible agents' behaviour, and therefore it must be the case that each norm can be eventually activated:

$$\forall ob \mathbf{EF}(state(ob) = activated) \tag{3}$$

If we assume that agents are autonomous, it should be possible for an agent both to violate and fulfill its obligations (and prohibitions), which means that norms regulate aspects of (social) reality which are contingent. It would be irrational to define an obligation characterized by a violation expression that makes it impossible to violate it:

$$\forall ob \mathbf{AF}(state(ob) = activated \rightarrow \mathbf{E} state(ob) = activated \mathbf{U} violated(ob)) \tag{4}$$

Similarly, we can specify that all obligations may be eventually fulfilled. Moreover, it should be the case that once a norm is activated, it ought to eventually reach a final state ($inactive$), which guarantees that the whole life-cycle of a norm is limited and regulated only by the institution that defines it:

$$\mathbf{AG} \forall ob(state(ob) = activated \rightarrow \mathbf{A}[state(ob) = activated \mathbf{U} state(ob) = inactive]) \tag{5}$$

Notice that any obligation which is not characterized by a deadline (not necessarily a time expression) violates property (5). For instance, if an agent ought to maintain indefinitely a state, as a consequence the whole institution would not satisfy property (5).

Obligations are defined to indicate whether a given behaviour is accepted by an institutions and typically "involve sacrifice and renunciation" [13]. For this reason, under the hypothesis that internal mental states are not accessible, it seems natural to assume that if an agent is obliged to a certain state of affairs it should not be provided with the possibility of revoking its own obligations. Otherwise, it could avoid complying with its duties without incurring in any sanction or blame. We formalize this fact as follows:

$$\begin{aligned} \mathbf{AG} \forall ob(state(ob) = activated \rightarrow \neg \mathbf{EX}(state(ob) = inactive \wedge \neg violation(ob) \\ \wedge \neg fulfillment(ob) \wedge \exists act\ done(act, liable(ob)))) \end{aligned} \tag{6}$$

The fact that an institution does not satisfy a domain-independent property does not necessarily mean that its rules prevent any successful interaction. It may be the case that all domain-dependent properties are satisfied while there are domain-independent properties that are violated. In any case, it is important that the designer becomes aware of this fact, and consider how to modify the institution. For instance, if property (1) does not hold, we can eliminate the institutional event or change powers and preconditions associated to it, otherwise its definition would be useless and the institution would be irrational.

4 A Symbolic Model Checker for Verifying Institutions

Figure 1 represents not only the metamodel of the institutional reality, but also it constitutes the abstract syntax of FIEVeL (*F*unctions for *I*nstitutionalized *E*nvironments *Ve*rification *L*anguage) [28, 29], a language that has been defined to *model* institutions in terms of the concepts introduced by our metamodel and to *verify* them by applying model-checking techniques [4]. As exemplified in Figure 3 and according to the institutional metamodel, an institution described in FIEVeL defines a set of status functions, characterized by powers and obligations, a set of institutional events and conventions for their performance. The semantics of FIEVeL constructs is given by describing how each expression affects the signature or the interpretation function of an OMSFOL logic characterized also by symbols introduced by our metamodel (see [29] for more details about FIEVeL).

The adoption of an order many-sorted first-order logic to describe the semantics of FIEVeL, its metamodel, and properties of institutions, naturally rises the question about how to verify OMSFOL models by applying model-checking techniques, which usually are applied to propositional models [4]. In the remainder of this section we will define an encoding E of OMSFOL models into propositional models and a translation τ of OMSFOL formulae into CTL* formulae which is exploited by our tool to verify institutions described in FIEVeL.

A propositional model is defined as a tuple $\widehat{\mathfrak{M}} = \left\langle \widehat{\mathfrak{F}}, \widehat{AP}, \widehat{V} \right\rangle$ where $\widehat{AP}$ is a set of atomic propositions, and $\widehat{V}$ is a valuation function which, given a state $\widehat{s} \in \widehat{\mathbf{S}}$ and a proposition $p \in \widehat{AP}$, returns a value in $\{0, 1\}$. Let be N_{D_σ} the cardinality of domain D_σ associated to sort σ: the set of atomic proposition $\widehat{AP}$ is determined by introducing a set of propositions $AP_{f_{x_1,..,x_n}} \in AP$ of cardinality $\log_2(N_{[\xi(f)]_0})$ for every function f and for every possible valuation, and a proposition $p_{x_1,...,x_n}$ for every predicate P and for every valuation. Moreover, for every domain we associate to each individual el a natural number n_{el} such that $0 \leq n_{el} < N_{D_\sigma}$. The encoding E of an OMSFOL model into a propositional model $\widehat{\mathfrak{M}}$ is defined as follows:

- each constant symbol c is encoded as a sequence of truth values corresponding to the binary representation of the number associated to the individual referenced by constant c ($E[c] = binary(I(c, s))$);
- each function f is encoded as the set of propositions induced by the current valuation ($E[f(x_1, ..., x_n)] = AP_{f(x_1,...,x_n)}$);
- each predicate P is encoded as the proposition induced by the current valuation ($E[P(x_1, ..., x_n)] = p_{x_1,...,x_n}$);

The valuation function $\widehat{V}$ is such that given an interpretation function I and a state s, propositions $AP_{f_{x_1,..,x_n}}$ are evaluated as the encoding of the element referred by $f(x_1, ..., x_n)$ and proposition $p_{x_1,...,x_n}$ evaluates as the truth value corresponding to predicate P in state s.

Assuming that the language is rich, that is, there exists a constant symbol for each individual of every domain, and also assuming for simplicity that only variables and constants can appear as arguments of functions and predicates, the translation of a OMSFOL formula φ with temporal operators is defined as follows:

- $\tau[t_1 = t_2] = \bigwedge_{i=0}^{i<N(\xi(t_1))} \neg(E(t_1)_i \oplus E(t_2)_i)$ where $E(t)_i$ refers to the i-th proposition (or equivalently the i-th truth value) corresponding to the encoding of term t;
- $\tau[P(x_1, ..., x_n)] = E(P(x_1, ..., x_n))$
- $\tau[\forall x\varphi] = \bigwedge_{c \in \mathbf{C}_{\xi(x)}} \tau[\varphi[c/x]]$ where c ranges over constants of sort $\xi(x)$ and $\varphi[c/x]$ is the result of replacing every free occurrence of x in φ with an occurrence of c;
- $\tau[\neg\varphi] = \neg\tau[\varphi]$
- $\tau[\varphi \wedge \varphi] = \tau[\varphi] \wedge \tau[\varphi]$
- $\tau[\mathbf{X}\varphi] = \mathbf{X}\tau[\varphi]$
- $\tau[\varphi \mathbf{U} \varphi] = \tau[\varphi] \mathbf{U} \tau[\varphi]$
- $\tau[\mathbf{E}\varphi] = \mathbf{E}\tau[\varphi]$

It can be demonstrated that:

Theorem 1 *Given a model $\mathfrak{M}$, characterized by a OMSFOL signature and finite domains, and an OMSFOL formula φ, $\mathfrak{M} \models \varphi$ if and only if $E[\mathfrak{M}] \models \tau[\varphi]$.*

To verify domain-dependent and domain-independent properties we have implemented a symbolic model checker based on the CUDD library [27]. Our tool takes as its input an institution modelled with FIEVeL, a list of individuals composing each basic domain, and a set of status functions which are assumed to be imposed at the first state. Domain-dependent properties are specified in a separate file, while domain-independent properties are selected by interacting with the tool. Starting from these inputs, the tool applies translation τ and encoding E to construct the corresponding Kripke structure, whose states and transition relation are symbolically represented as OBDDs [3]. To verify properties whose translation corresponds to CTL formulae, that is, formulae where path quantifiers ($\mathbf{A}$ and $\mathbf{E}$) are immediately followed by temporal operators ($\mathbf{X}$, $\mathbf{F}$, and $\mathbf{G}$), the tool applies symbolic algorithms as described in [4].

```
basic-types:
   priceD subtype-of INT;;
     ...
base-events :
 message refuse(buyer:AID,good:OID,value:priceD);
     ...
institution purchase {
  ...
    status-function proposed(proposer:AID,good:OID,value:priceD) {
     key proposer,good
     powers
     ...
      refuseRequest d <- TRUE;
     ...
     deontic:
      p1 obligation(TRUE,
         done(transferPossession,subject)),activation>1);
     ...
   }// proposed
     ...
  institutional-events:
     ...
     institutional-action refuseRequest(buyer:AID,g:OID,price:priceD)
     pre exists p in proposed (((p.subject=actor and
      and (p.proposer=buyer and p.value=price)));
     eff revoke proposed (subject=actor,proposer=buyer,good=g, value=price);
     ...
  conventions:
     exch-Msg(refuse) p [TRUE] =c=> refuseRequest
     [buyer =c=> buyer
     good =c=> g
     value =c=> price ];
     ...
}// institution
```

Figure 3: Fragments of the institution of property.

5 Modelling and Verifying the Institution of Property

In this section we report the results obtained during the verification process of the *institution of property* as it has been defined in [5]. During the formalization of such institution (see Figure 3) we have slightly modified certain features of the original specification to adapt them to our concepts. Despite these changes, results obtained with our formalization can be extended to the institutions described in [5]. For the sake of brevity we will focus on those aspects that our analysis has shown to be problematic and which we deem highlight the importance of the definition of a metamodel of institutions.

In [5] the institution of property defines an institutional action, *transfer_ownership*, which is "empowered if the initiator of the transfer is the *owner* of the object being transferred". Furthermore a set of institutional actions are introduced, like *requestGoods* which allows a *customer* to request a good, *sendPayment* which can be performed by a customer to send a payment, *refuseRequest* which allows the *merchant* to refuse a request, and finally *sendGoods* which empowers the merchant to send a good. Notice that in [5] it is not required that institutional actions happen because agents have performed basic actions conventionally bounded to them, which obliged us to introduce a set of base-level events and conventions to enable agents to perform institutional actions described above.

According to [5] "sending a request for goods creates an obligation on the merchant to have sent the goods before the interaction ends" but "sending the refusal cancels the merchant's obligation to send goods". Therefore, the merchant can cancel an obligation of itself which intuitively violates property (6) discussed in the previous section.

Let us now consider how agents can execute action *transfer_good*, which is defined in such a way that it can be successfully performed at time t if and only if either the merchant sends the good at time t and the customer has payed at time $t_1 < t$, or the customer pays at time t and the merchant has sent the good at time $t_1 < t$. Given such description it seems natural to introduce two conventions: $conv_1$ which states that the performance of the institutional act $sendGood$ counts as $transfer\_property$ if the customer has payed, and $conv_2$ such that $sendPayment$ counts as $tranfer\_property$ if the merchant has sent the good. Such conventions reflect the fact that in [5] the execution of sending a good (or similarly sending a

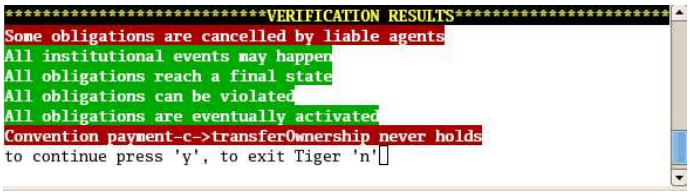

Figure 4: The report generated by our tool during the verification of domain-independent properties discussed in Section 3.

payment) implies the transfer of property. At this point it is important to notice that if $transfer\_property$, $sendGoods$, and $sendPayment$ are institutional actions, then they are successfully executed only if the actor is empowered (see axiom (5) which is similar to an inference rule in [5]). The concept of power adopted by [5] is similar to the one presented here, since it is defined as "the capability of an agent to bring about a change" in the institutional state. But, the act of paying is performed by the customer, while the transfer of property is empowered to the merchant, and they cannot be the same agent in a given transaction. These considerations rise a problem of agency of the institutional actions: how can a payment of agent a count as an action performed by agent b? It would be possible if agent b had delegated its power to agent a, but the authors do not introduce delegation of powers in their formalization. Therefore, we should expect that property (2) does not hold, since according to axiom (6) the convention among the payment and the transfer of property can never produce the successful performance of both acts.

Figure 4 shows the verification results generated by our tool when it has been asked to verify properties described in Section 3 and where we can observe that properties (2) and (6) are violated. The generation of the transition system corresponding to the institution of property and the verification of all domain-independent properties required 0.25 seconds on a laptop with installed Linux and equipped with a pentium 1.66 GHz and 1 GB of RAM.

When a domain-independent property is not satisfied by an institution, we can consider what specific features of the metamodel it reflects to get some clue about how to modify an institution in order to satisfy it. For instance, we know that all institutional events may eventually happen as ensured by property (1). Therefore, as we have also noted in the previous informal analysis, a way to satisfy property (2) is to change the institutional power of agents. In particular, here we propose to not classify the exchange of the property as an action but as an institutional event. This solution immediately solves the agency problem and eliminates the power attributed to the owner of a good.

To satisfy property (6), that is, agents cannot cancel their own obligations, we propose to model the act of requesting a good in such a way that it does not create an obligation for the merchant to send the good, but it creates an obligation to agree or reject the request of the customer. In both cases, an answer of the merchant satisfies its obligation; in particular, a positive answer imposes on the involved agents a set of status functions which represent the obligation to transfer the possession (or perform the payment) and the powers to do so. Notice that such a formalization better represents the fact that a directive act (like a request) does not create obligations to the performance of the requested act [24], while a commissive (like the agree communicative act) commits the agent to the performance of the agreed action (see also [9]).

6 Discussion and Conclusions

In this paper we have presented a metamodel for verifying institutional reality based on the notion of status function, which is regarded as a (possibly empty) aggregate of deontic positions (powers, obligation, etc.). This approach provides for a unified view of institutional facts and deontic positions, which have been usually analyzed separately [5, 9, 22], and is motivated by the fact that institutional reality is such only because it is constituted by deontic positions attributed to agents. We have introduced the semantics of our institutional metamodel in terms of an order many-sorted first-order logic, which allows us to formalize both axioms regulating the metamodel and a set of domain-independent properties which reflect the intended meaning of concepts defined by our metamodel. Finally, we have tested our approach by verifying the institution presented in [5] with respect to a set of domain-independent properties and shown that the verification of such properties enhances the formalization of sound institutions.

In the literature there are several attempts to *model* and *verify* institutions and normative systems. In [9]

the authors rely on an intuitive semantics to model institutional reality in terms of entities, roles, and norms. Instead, the approach presented in this paper provides a formal semantics of institutional concepts such that we can apply model checking techniques to verify institutions.

In [22] normative systems are described by using the Event Calculus [18]. The absence of an institutional metamodel, which for instance provides an axiom to state that every institutional action must be authorized in order to be successfully executed, obliges the authors of [22] to specify this fact for every single action and for every role. Therefore, the definition of a metamodel provides a significant advantage, especially when many status functions (or roles, using the terminology of [22]) are authorized to perform the same institutional action. Furthermore, the definition of an encoding of institutions described in FIEVeL into propositional models allows us to verify our systems, while in [22] the authors must rely on "systematic runs".

In [15] the authors propose a framework to model check *electronic institutions*, a formalism proposed in [7] and which describes institutions as finite automata. Starting from this point, in [15] the authors limit their attention on the verification of properties of finite automata (e.g. "it is always possible to reach a final state"). It is important to notice that in [15] arcs of electronic institutions (which in principle represent permitted acts [7]) are interpreted as obligatory moves of agents, which may lead the model checker to answer that a given property holds in an institution while it is not the case and vice versa.

We plan to extend our metamodel, and consequently our modelling language, to model different interdependent institutions like in [30], which raises, among others, two interesting research problems: first, how to model interdependencies among different contexts, and second, how to design an institution which somehow depends on another institution.

Acknowledgments This research has been supported by Swiss National Science Foundation project 200020-109525, "Artificial Institutions: specification and verification of open distributed interaction frameworks". The author would like to thank his Ph.D. advisor, Marco Colombetti, for fruitful discussions and criticisms about the contents presented in this paper and Alessio Lomuscio for his advices regarding the implementation of the tool discussed in the paper.

References

[1] P. Blackburn, M. de Rijke, and Y. Venema. *Modal Logic*. Cambridge University Press, 2001.

[2] G. Boella and L. van der Torre. The ontological properties of social roles: Definitional dependence, powers and roles playing roles. In *Proceedings of the ICAIL05 Workshop on Legal Ontologies and Artificial Intelligence Techniques*, 2005.

[3] R. E. Bryant. Graph-based algorithms for boolean function manipulation. *IEEE Trans. Comput.*, 35(8):677–691, 1986.

[4] E. M. Clarke, O. Grumberg, and D. Peled. *Model Checking*. MIT Press, 1999.

[5] O. Cliffe, M. D. Vos, and J. Padget. Specifying and Analysing Agent-based Social Institutions using Answer Set Programming. In *Coordination, Organization, Institutions and Norms in Agent Systems I*, number 3913 in LNCS, pages 99–113, 2005.

[6] M. Esteva, J. A. Rodríguez-Aguilar, B. Rosell, and J. L. Arcos. AMELI: An Agent-based Middleware for Electronic Institutions. In N. R. Jennings, C. Sierra, L. Sonenberg, and M. Tambe, editors, *Proceedings of the 3rd International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS 2004)*, pages 236–243. ACM Press, 2004.

[7] M. Esteva, J. A. Rodríguez-Aguilar, C. Sierra, P. Garcia, and J. L. Arcos. On the Formal Specification of Electronic Institutions. In *Agent Mediated Electronic Commerce, The European AgentLink Perspective*, volume 1991 of *LNAI*, pages 126–147, 2001.

[8] A. D. H. Farrell, M. J. Sergot, M. Sallé, and C. Bartolini. Using the Event Calculus for Tracking the Normative State of Contracts. *Journal of Cooperative Information Systems*, 14(2-3):99–129, 2005.

[9] N. Fornara, F. Viganò, and M. Colombetti. Agent Communication and Institutional Reality. In *Agent Communication*, volume 3396 of *LNAI*, pages 1–17, 2005.

[10] A. Garcia-Camino, P. Noriega, and J. A. Rodríguez-Aguilar. Implementing norms in electronic institutions. In *Proceedings of the 4th International Joint Conference on Autonomous agents and Multi-Agent Systems*, pages 667–673, 2005.

[11] A. García-Camino, J. A. Rodríguez-Aguilar, C. Sierra, and W. W. Vasconcelos. A distributed architecture for norm-aware agent societies. In *Declarative Agent Languages and Technologies III (DALT 2005*, volume 3904 of *LNCS*, pages 89–105. Springer, 2005.

[12] D. Grossi, J.-J. C. Meyer, and F. Dignum. Counts-as: Classification or constitution? an answer using modal logic. In *Proceedings of the 8th International Workshop on Deontic Logic in Computer Science*, volume 4048 of *LNCS*, pages 115–130, 2006.

[13] H. L. A. Hart. *The Concept of Law*. Oxford University Press, 1961.

[14] G. Holzmann. *The SPIN Model Checker: Primer and Reference Manual*. Addison Wesley, 2003.

[15] M.-P. Huget, M. Esteva, S. Phelps, C. Sierra, and M. Wooldridge. Model Checking Electronic Institutions. In *Proceedings of the ECAI Workshop on Model Checking and Artificial Intelligence (MoChArt I)*, 2002.

[16] A. Jones and M. J. Sergot. A formal characterisation of institutionalised power. *Journal of the IGPL*, 4(3):429–445, 1996.

[17] A. Kleppe, J. Warmer, and W. Bast. *MDA Explained: The Model Driven Architecture–Practice and Promise*. Addison-Wesley Professional, Reading, Massachusetts, USA, 1 edition, 2003.

[18] R. A. Kowalski and M. J. Sergot. A Logic-based Calculus of Events. *New Generation Computing*, 4:67–95, 1986.

[19] A. Lomuscio and M. Sergot. A formulation of violation, error recovery, and enforcement in the bit transmission problem. *Journal of Applied Logic (Selected articles from DEON02 - London)*, 1(2):93–116, 2002.

[20] K. Meinke and J. V. Tucker, editors. *Many-sorted logic and its applications*. John Wiley & Sons, Inc., 1993.

[21] J. S. Ostroff. Deciding properties of timed transition models. *IEEE Transactions on Parallel Distributed Systems*, 1(2):170–183, 1990.

[22] J. Pitt, L. Kamara, M. Sergot, and A. Artikis. Formalization of a voting protocol for virtual organizations. In *Proceedings of the 4th International Joint Conference on Autonomous agents and Multi-Agent Systems (AAMAS 2005)*, pages 373–380, 2005.

[23] F. Raimondi and A. Lomuscio. Automatic verification of deontic interpreted systems by model checking via OBDDs. *Journal of Applied Logic*. To appear.

[24] J. R. Searle. *Speech Acts: An Essay in the Philosophy of Language*. Cambridge University Press, 1969.

[25] J. R. Searle. *The construction of social reality*. Free Press, 1995.

[26] M. J. Sergot and R. Craven. The Deontic Component of Action Language nC+. In *Proceedings of the 8th International Workshop on Deontic Logic in Computer Science*, volume 4048 of *LNCS*, pages 222–237, 2006.

[27] F. Somenzi. Cudd: Cu decision diagram package. http://vlsi.colorado.edu/ fabio/CUDD/.

[28] F. Viganò. A Framework for Model Checking Institutions. In *Proceedings of the ECAI Workshop on Model checking and Artificial Intelligence (MoChArt IV)*, 2006.

[29] F. Viganò. FIEVeL, a Language for the Specification and Verification of Institutions. Technical Report 3, Institute for Communication Technologies, Università della Svizzera Italiana, 2006.

[30] F. Viganò, N. Fornara, and M. Colombetti. An Event Driven Approach to Norms in Artificial Institutions. In *Coordination, Organization, Institutions and Norms in Multi-Agent Systems*, volume 3913 of *LNAI*, pages 142–154. Springer, 2006.