

# A Formal Road from Institutional Norms to Organizational Structures

#331\*

## ABSTRACT

Up to now, the way institutions and organizations have been used in the development of open systems has not often gone further than a useful heuristics. In order to develop systems actually implementing institutions and organizations, formal methods should take the place of heuristic ones. The paper presents a formal semantics for the notion of institution and its components (abstract and concrete norms, empowerment of agents, roles) and defines a formal relation between institutions and organizational structures. As a result, it is shown how institutional norms can be refined to constructs—organizational structures—which are closer to an implemented system. It is also shown how such a refinement process can be fully formalized and it is therefore amenable to rigorous verification.

## 1. INTRODUCTION

The opportunity of a “technology transfer” from the field of organizational and social theory to distributed AI and multiagent systems (MASs) has long been advocated ([8]). In MASs the application of the organizational and institutional metaphors to system design has proven to be useful for the development of methodologies and tools. In many cases, however, the application of these conceptual apparatuses amounts to mere heuristics guiding the high level design of the systems. It is our thesis that the application of those apparatuses can be pushed further once their key concepts are treated formally, that is, once notions such as norm, role, structure, etc. obtain a formal semantics. This has been the case for agent programming languages after the relevant concepts borrowed from folk psychology (belief, intention, desire, knowledge, etc.) have been addressed in comprehensive formal logical theories such as, for instance, BDICTL ([22]) and KARO ([17]). As a matter of fact, those theories have fostered the production of architectures and programming languages.

---

\*The primary author of this paper is a student.

What is lacking at the moment for the design and development of open MASs is, in our opinion, something that can play the role that BDI-like formalisms have played for the design and development of single-agent architectures. Aim of the present paper is to fill this gap with respect to the notion of institution providing formal foundations for the application of the institutional metaphor and for its relation to the organizational one. The main result of the paper consists in showing how abstract constraints (institutions) can be step by step refined to concrete structural descriptions (organizational structures) of the to-be-implemented system, bridging thus the gap between abstract norms and concrete system specifications.

Concretely, in Section 2, a logical framework is presented which provides a formal semantics for the notions of institution, norm, role, and which supports the account of key features of institutions such as the translation of abstract norms into concrete and implementable ones, the institutional empowerment of agents, and some aspects of the design of norm enforcement. In Section 3 the framework is extended to deal with the notion of the infrastructure of an institution. The extended framework is then studied in relation to the formalism for representing organizational structures presented in [11]. In Section 4 some conclusions follow.

## 2. INSTITUTIONS

Social theory usually thinks of institutions as “the rules of the game” ([18, 23]). From an agent perspective institutions are, to paraphrase this quote, “the rules of the various games agents can play in order to interact with one another”. To assume an institutional perspective on MAS means therefore to think of MAS in normative terms:

[...] law, computer systems, and many other kinds of organizational structure may be viewed as instances of *normative systems*. We use the term to refer to any set of interacting agents whose behavior can usefully be regarded as governed by norms ([15], p.276).

The normative system perspective on institutions is, as such, nothing original and it is already a quite acknowledged position within the community working on electronic institutions, or eInstitutions ([26]). What has not been sufficiently investigated and understood with formal methods is, in our view, the question: what does it amount to, for a MAS, to be put under a set of norms? Or in other words: what does it mean for a designer of an eInstitution to state a set of norms? We advance a precise thesis on this issue, which is

also inspired by work in social theory:

Now, as the original manner of producing physical entities is creation, there is hardly a better way to describe the production of moral entities than by the word ‘*imposition*’ [impositio]. For moral entities do not arise from the intrinsic substantial principles of things but *are superadded to things already existent and physically complete* ([21], pp. 100-101).

By ignoring for a second the philosophical jargon of the Seventeenth century we can easily extract an illuminating message from the excerpt: what institutions do is to impose properties on already existing entities. That is to say, institutions provide descriptions of entities by making use of conceptualizations that are not proper of the common descriptions of those entities. For example, that cars have wheels is a common factual property, whereas the fact that cars count as vehicles in some technical legal sense is a property that law imposes on the concept “car”. To say it with [25], the fact that cars have wheels is a *brute fact*, while the fact that cars are vehicles is an *institutional fact*. Institutions build structured descriptions of institutional properties upon brute descriptions of a given domain.

At this point, the step toward eInstitutions is natural. eInstitutions impose properties on the possible states of a MAS: they specify what are the states in which an agent  $i$  enacts a role  $r$ ; what are the states in which a certain agent is violating the norms of the institution, etc. They do this via linking some institutional properties of the possible states and transitions of the system (e.g., agent  $i$  enacts role  $r$ ) to some brute properties of those states and transitions (e.g., agent  $i$  performs protocol No.56). An institutional property is therefore a property of system states or system transitions (i.e., a state type or a transition type) that does not belong to a merely technical, or factual, description of the system.

To sum up, institution are viewed as sets of norms (normative system perspective), and norms are thought of as the imposition of an institutional description of the system upon its description in terms of brute properties. In a nutshell, *institutions are impositions of institutional terminologies upon brute ones*. The following sections provide a formal analysis of this thesis and show its explanatory power in delivering a rigorous understanding of key features of institutions. Because of its suitability for representing complex domain descriptions, the formal framework we will make use of is the one of Description Logics (DL). The use of such formalism will also stress the idea of viewing institutions as the impositions of domain descriptions.

## 2.1 Preliminaries: a very expressive DL

The description logic language enabling the necessary expressivity expands the standard description logic language  $\mathcal{ALC}$  ([3]) with relational operators ( $\sqcup, \circ, \neg, id$ ) to express complex transition types, and relational hierarchies ( $\mathcal{H}$ ) to express inclusion between transition types. Following a notational convention common withing DL we denote this language with  $\mathcal{ALCH}^{(\sqcup, \circ, \neg, id)}$ .

DEFINITION 1. (*Syntax of  $\mathcal{ALCH}^{(\sqcup, \circ, \neg, id)}$* ) *transition types and state type constructs are defined by the*

following BNF:

$$\begin{aligned} \alpha &:= a \mid \alpha \circ \alpha \mid \alpha \sqcup \alpha \mid \neg \alpha \mid id(\gamma) \\ \gamma &:= c \mid \perp \mid \neg \gamma \mid \gamma \sqcap \gamma \mid \forall \alpha. \gamma \end{aligned}$$

It is worth providing the intuitive reading of a couple of the operators and the constructs just introduced. In particular  $\forall \alpha. \gamma$  has to be read as: “after all executions of transitions of type  $\alpha$ , states of type  $\gamma$  are reached”. The operator  $\circ$  denotes the concatenation of transition types. The operator  $id$  applies to a state description  $\gamma$  and yields a transition description, namely, the transition ending in  $\gamma$  states. It is the description logic variant of the test operator in Dynamic Logic ([5]). Notice that we use the same symbols  $\sqcup$  and  $\neg$  for denoting the boolean operators of disjunction and negation of both state and transition types.

A terminological box (henceforth TBox)  $T = \langle \Gamma, A \rangle$  consists of a finite set  $\Gamma$  of state type inclusion assertions ( $\gamma_1 \sqsubseteq \gamma_2$ ), and of a finite set  $A$  of transition type inclusion assertions ( $\alpha_1 \sqsubseteq \alpha_2$ ).

The semantics of  $\mathcal{ALCH}^{(\sqcup, \circ, \neg, id)}$  is model theoretical and it is given in terms of interpreted transition systems. As usual, state types are interpreted as sets of states and transition types as sets of state pairs.

DEFINITION 2. (*Semantics of  $\mathcal{ALCH}^{(\sqcup, \circ, \neg, id)}$* )

*An interpreted transition system  $m$  for  $\mathcal{ALCH}^{(\sqcup, \circ, \neg, id)}$  is a structure  $\langle S, \mathcal{I} \rangle$  where  $S$  is a non-empty set of states and  $\mathcal{I}$  is a function such that:*

$$\begin{aligned} \mathcal{I}(c) &\subseteq S & \mathcal{I}(a) &\subseteq S \times S \\ \mathcal{I}(\perp) &= \emptyset & \mathcal{I}(\neg \gamma) &= \Delta_m \setminus \mathcal{I}(\gamma) \\ \mathcal{I}(\gamma_1 \sqcap \gamma_2) &= \mathcal{I}(\gamma_1) \cap \mathcal{I}(\gamma_2) \\ \mathcal{I}(\forall \alpha. \gamma) &= \{s \in S \mid \forall t, (s, t) \in \mathcal{I}(\alpha) \Rightarrow t \in \mathcal{I}(\gamma)\} \\ \mathcal{I}(\alpha_1 \sqcup \alpha_2) &= \mathcal{I}(\alpha_1) \cup \mathcal{I}(\alpha_2) \\ \mathcal{I}(\neg \alpha) &= S \times S \setminus \mathcal{I}(\alpha) \\ \mathcal{I}(\alpha_1 \circ \alpha_2) &= \{(s, s') \mid \exists s'', (s, s'') \in \mathcal{I}(\alpha_1) \ \& \ (s'', s') \in \mathcal{I}(\alpha_2)\} \\ \mathcal{I}(id(\gamma)) &= \{(s, s) \mid s \in \mathcal{I}(\gamma)\} \end{aligned}$$

*An interpreted transition system  $m$  is a model of a state type inclusion assertion  $\gamma_1 \sqsubseteq \gamma_2$  if  $\mathcal{I}(\gamma_1) \subseteq \mathcal{I}(\gamma_2)$ . It is a model of a transition type inclusion assertion  $\alpha_1 \sqsubseteq \alpha_2$  if  $\mathcal{I}(\alpha_1) \subseteq \mathcal{I}(\alpha_2)$ . An interpreted transition system  $m$  is a model of a TBox  $T = \langle \Gamma, A \rangle$  if  $m$  is a model of each inclusion assertion in  $\Gamma$  and  $A$ .*

REMARK 1. (*Derived constructs*) *The correspondence between description logic and dynamic logic is well-known ([3]). In fact, the language presented in Definitions 1 and 2 is a notational variant of the language of Dynamic Logic ([5]) without the iteration operator of transition types. As a consequence, some key constructs are definable in  $\mathcal{ALCH}^{(\sqcup, \circ, \neg, id)}$ . In particular we will make use of the following definition of the if-then-else transition type:*

$$\text{if } \gamma \text{ then } \alpha_1 \text{ else } \alpha_2 = (id(\gamma) \circ \alpha_1) \sqcup (id(\neg \gamma) \circ \alpha_2).$$

*Boolean operators are defined as usual.*

We will come back to some complexity features of this logic in Section 2.5.

## 2.2 Institutions as terminologies

We have upheld that institutions “impose” new system descriptions which are formulated in terms of sets of norms.

The step toward a formal grounding of this view of institutions is now short: norms can be thought of as terminological axioms, and institutions as sets of terminological axioms, i.e., terminological boxes.

An institution can be specified as a terminological box  $\mathcal{I}ns = \langle \Gamma_{ins}, A_{ins} \rangle$ , where each inclusion statement in  $\Gamma_{ins}$  and  $A_{ins}$  models a norm of the institution. Obviously, not every TBox can be considered to be an institution specification. In particular, an institution specification  $\mathcal{I}ns$  must have some precise linguistic relationship with the ‘brute’ descriptions upon which the institution is specified. We denote by  $\mathcal{L}_{ins}$  the non-logical alphabet containing only institutional state and transition types, and by  $\mathcal{L}_{brute}$  the non-logical alphabet containing those types taken to talk about, instead, ‘brute’ states and transitions<sup>1</sup>.

DEFINITION 3. (*Institutions as TBoxes*)

A TBox  $\mathcal{I}ns = \langle \Gamma_{ins}, A_{ins} \rangle$  is an institution specification if:

1. The non-logical alphabet on which  $\mathcal{I}ns$  is specified contains elements of both  $\mathcal{L}_{ins}$  and  $\mathcal{L}_{brute}$ . In symbols:  $\mathcal{L}(\mathcal{I}ns) \subseteq \mathcal{L}_{ins} \cup \mathcal{L}_{brute}$ .
2. There exist sets of terminological axioms  $\Gamma_{bridge} \subseteq \Gamma_{ins}$  and  $A_{bridge} \subseteq A_{ins}$  such that either the left-hand side of these axioms is always a description expressed in  $\mathcal{L}_{brute}$  and the right-hand side a description expressed in  $\mathcal{L}_{ins}$ , or those axioms are definitions. In symbols: if  $\gamma_1 \sqsubseteq \gamma_2 \in \Gamma_{bridge}$  then either  $\gamma_1 \in \mathcal{L}_{brute}$  and  $\gamma_2 \in \mathcal{L}_{ins}$  or it is the case that also  $\gamma_2 \sqsubseteq \gamma_1 \in \Gamma_{bridge}$ . The clause for  $A_{bridge}$  is analogous.
3. The remaining sets of terminological axioms  $\Gamma_{ins} \setminus \Gamma_{bridge}$  and  $A_{ins} \setminus A_{bridge}$  are all expressed in  $\mathcal{L}_{ins}$ . In symbols:  $\mathcal{L}(\Gamma_{ins} \setminus \Gamma_{bridge}) \subseteq \mathcal{L}_{ins}$  and  $\mathcal{L}(A_{ins} \setminus A_{bridge}) \subseteq \mathcal{L}_{ins}$ .

The definition states that an institution specification needs to be expressed on a language including institutional as well as brute terms (1); that a part of the specification concerns a description of mere institutional terms (3); and that there needs to be a part of the specification which connects institutional terms to brute ones (2). Terminological axioms in  $\Gamma_{bridge}$  and  $A_{bridge}$  formalize in DL the Searlean notion of counts-as conditional ([25]), that is, rules stating what kind of meaning an institution gives to certain brute facts and transitions (e.g., checking box No.4 in form No.2 counts as accepting your personal data to be used for research purposes). A formal theory of counts-as statements has been thoroughly developed in a series of papers among which [10, 13]. The technical content of the present paper heavily capitalizes on that work.

Notice also that given the semantics presented in Definition 2, if institutions can be specified via TBoxes then the meaning of such specifications is a set of interpreted transition systems, i.e., the models of those TBoxes. These transition systems can be in turn thought of as all the possible MASs which model the specified institution.

REMARK 2. ( *$\mathcal{L}_{brute}$  from a designer’s perspective*) From a design perspective language  $\mathcal{L}_{brute}$  has to be thought of as the language on which a designer would specify a system instantiating a given institution<sup>2</sup>. Definition 3 shows that

<sup>1</sup>Symbols from  $\mathcal{L}_{ins}$  and  $\mathcal{L}_{brute}$  will be indexed (especially with agent identifiers) to add some syntactic sugar.

<sup>2</sup>To make a concrete example, the AMELI middleware [7] can be viewed as a specification tool at a  $\mathcal{L}_{brute}$  level.

for such a design task it is needed to formally specify an explicit bridge between the concepts used in the description of the actual system and the institutional ‘abstract’ concepts. We will come back to this issue in Section 3.

## 2.3 From abstract to concrete norms

To illustrate Definition 3, and show its explanatory power, an example follows which depicts an essential phenomenon of institutions.

EXAMPLE 1. (*From abstract to concrete norms*) Consider an institution supposed to regulate access to a set of public web services. It may contain the following norm: “it is forbidden to discriminate access on the basis of citizenship”. Suppose now a system has to be built which complies with this norm. The first question is: what does it mean, in concrete, “to discriminate on the basis of citizenship”? The system designer should make some concrete choices for interpreting the norm and these choices should be kept track of in order to explicitly link the abstract norm to its concrete interpretation. The problem can be represented as follows. The abstract norm is formalized by Formula 1 by making use of a standard reduction technique of deontic notions (see [16]): the statement “it is forbidden to discriminate on the basis of citizenship” amounts to the statement “after every execution of a transition of type  $DISCR(i, j)$  the system always ends up in a violation state”. Together with the norm also some intuitive background knowledge about the discrimination action needs to be formalized. Here, as well as in the rest of the examples in the paper, we provide just that part of the formalization which is strictly functional to show how the formalism works in practice. Formulae 2 and 3 express two effect laws: if the requester  $j$  is Dutch then after all executions of transitions of type  $DISCR(i, j)$   $j$  is accepted by  $i$ , whereas if it is not all the executions of the transitions of the same type have as an effect that it is not accepted.

$$\forall DISCR(i, j).viol \equiv \top \quad (1)$$

$$dutch(j) \sqsubseteq \forall DISCR(i, j).accepted(j) \quad (2)$$

$$\neg dutch(j) \sqsubseteq \forall DISCR(i, j).\neg accepted(j) \quad (3)$$

The rest of the axioms concern the translation of the abstract type  $DISCR(i, j)$  to concrete transition types. Formula 4 refines it by making explicit that a precise if-then-else procedure counts as a discriminatory act of agent  $i$ . Formulae 5 and 6 specify which messages of  $i$  to  $j$  count as acceptance and rejection. If the designer uses transition types  $SEND(msg33, i, j)$  and  $SEND(msg38, i, j)$  for the concrete system specification, then Formulae 5 and 6 can be thought of as bridge axioms connecting notions belonging to the institutional alphabet (to accept, and to reject) to concrete ones (to send specific messages). Finally, Formulae 7 and 8 state two intuitive effect laws concerning the  $ACCEPT(i, j)$  and  $REJECT(i, j)$  types.

$$\text{if } dutch(j) \text{ then } ACCEPT(i, j) \quad (4)$$

$$\text{else } REJECT(i, j) \sqsubseteq DISCR(i, j) \quad (5)$$

$$SEND(msg33, i, j) \sqsubseteq ACCEPT(i, j) \quad (6)$$

$$SEND(msg38, i, j) \sqsubseteq REJECT(i, j) \quad (7)$$

$$\forall ACCEPT(i, j).accepted(j) \equiv \top \quad (8)$$

$$\forall REJECT(i, j).\neg accepted(j) \equiv \top \quad (8)$$

It is easy to see, on the grounds of the semantics exposed in Definition 2, that the following concrete inclusion statement

holds w.r.t. the specified institution:

$$\begin{aligned} & \text{if } \text{dutch}(j) \text{ then } \text{SEND}(\text{msg33}, i, j) \\ & \text{else } \text{SEND}(\text{msg38}, i, j) \sqsubseteq \text{DISCR}(i, j) \end{aligned} \quad (9)$$

This scenario exemplifies a pervasive feature of human institutions which, as extensively argued in [10], should be incorporated by electronic ones. Current formal approaches to institutions, such as ISLANDER [6], do not allow for the formal specification of explicit translations of abstract norms into concrete ones, and focus only on norms that can be specified at the concrete system specification level. What Example 1 shows is that the problem of the abstractness of norms in institutions can be formally addressed and can be given a precise formal semantics.

The scenario suggests that, just by modifying an appropriate set of terminological axioms, it is possible for the designer to obtain a different institution by just modifying the sets of bridge axioms without touching the terminological axioms expressed only in the institutional language  $\mathcal{L}_{ins}$ . In fact, it is the case that a same set of abstract norms can be translated to different and even incompatible sets of concrete norms. This translation can nevertheless not be arbitrary ([1]).

**EXAMPLE 2.** (*Acceptable and unacceptable translations of abstract norms*) Reconsider again the scenario sketched in Example 1. The transition type  $\text{DISCR}(i, j)$  has been translated to a complex procedure composed by concrete transition types. Would any translation do? Consider an alternative institution specification  $\mathfrak{Ins}'$  containing Formulae 1-3 and the following translation rule:

$$\text{PAY}(j, i, \text{€}10) \sqsubseteq \text{DISCR}(i, j) \quad (10)$$

Would this formula be an acceptable translation of the abstract norm expressed in Formula 1? The axiom states that transitions where  $j$  receives  $\text{€}10$  from  $i$  count as transitions of type  $\text{DISCR}(i, j)$ . Needless to say this is not intuitive, because the abstract transition type  $\text{DISCR}(i, j)$  obeys some intuitive conceptual constraints (Formulae 2 and 3) that all its translations should also obey. In fact, the following inclusions would then hold in  $\mathfrak{Ins}'$ :

$$\text{dutch}(j) \sqsubseteq \forall \text{PAY}(j, i, \text{€}10).\text{accepted}(j) \quad (11)$$

$$\neg \text{dutch}(j) \sqsubseteq \forall \text{PAY}(j, i, \text{€}10).\neg \text{accepted}(j) \quad (12)$$

In fact, such state properties for the formalization of the transition type  $\text{PAY}(j, i, \text{€}10)$  are intuitively unacceptable. A transition type  $\text{PAY}(j, i, \text{€}10)$  should not yield the effects that a sub-type of  $\text{DISCR}(i, j)$  yields. It is on the contrary perfectly intuitive that Formula 9 obeys the constraints in Formulae 2 and 3, which it does, as it can be easily checked on the grounds of the semantics.

It is worth stressing that without providing a set theoretic semantics for the translation rules linking the institutional notions to the brute ones, it would not be so straightforward to model the logical constraints to which the translations are subjected (Example 2). This is precisely the advantage of viewing translation rules as specific terminological axioms, i.e.,  $\Gamma_{\text{bridge}}$  and  $A_{\text{bridge}}$ , working as a bridge between two languages (Definition 3). In [12], we have thoroughly compared this approach with approaches such as [9] which conceive of translation rules as inference rules.

The two examples have shown how our approach can account for some essential features of institutions. In the next section the same framework is applied to provide a formal analysis of the notion of role.

## 2.4 Institutional modules and roles

Viewing institutions as the impositions of institutional descriptions on systems' states and transitions allows for analyzing the normative system perspective itself (i.e., institutions are sets of norms) at a finer granularity. We have seen that the terminological axioms specifying an institution concern complex descriptions of new institutional notions. Some of the institutional state types occurring in the institution specification play a key role in structuring the specification of the institution itself. The paradigmatic example in this sense ([25]) are facts such as “agent  $i$  enacts role  $r$ ” which will be denoted by state types  $\text{rea}(i, r)$ . By stating how an agent can enact and ‘deact’ a role  $r$ , and what normative consequences follow from the enactment of  $r$ , an institution describes expected forms of agents' behavior while at the same time abstracting from the concrete agents taking part to the system.

The sets of norms specifying an institution can be clustered on the grounds of the  $\text{rea}$  state types. For each relevant institutional state type (e.g.,  $\text{rea}(i, r)$ ), the terminological axioms which define an institution, i.e., its norms, can be clustered in (possibly overlapping) sets of three different types: the axioms specifying how states of that institutional type can be reached (e.g., how an agent  $i$  can enact the role  $r$ ); how states of that type can be left (e.g., how an agent  $i$  can ‘deact’ the a role  $r$ ); and what kind of institutional consequences do those states bear (e.g., what rights and power does agent  $i$  acquire by enacting role  $r$ ). Borrowing the terminology from work in legal and institutional theory ([23, 25]), these clusters of norms can be called, respectively, *institutive*, *terminative* and *status* modules.

**Status modules** We call status modules those sets of terminological axioms which specify the institutional consequences of the occurrence of a given institutional state-of-affairs, for instance, the fact that agent  $i$  enacts role  $r$ .

**EXAMPLE 3.** (*A status module for roles*) Enacting a role within an institution bears some institutional consequences that are grouped under the notion of status: by playing a role an agent acquires a specific status. Some of these consequences are deontic and concern the obligations, rights, permissions under which the agent puts itself once it enacts the role. An example which pertains to the normative description of the status of both a “buyer” and a “seller” roles is the following:

$$\begin{aligned} & \text{rea}(i, \text{buyer}) \sqcap \text{rea}(j, \text{seller}) \sqcap \text{win\_bid}(i, j, b) \\ & \qquad \qquad \qquad \sqsubseteq \forall \neg \text{PAY}(i, j, b).\text{viol}(i) \end{aligned} \quad (13)$$

If agent  $i$  enacts the buyer role and  $j$  the seller role and  $i$  wins bid  $b$  then if  $i$  does not perform a transition of type  $\text{PAY}(i, j, b)$ , i.e., does not pay to  $j$  the price corresponding to bid  $b$ , then the system ends up in a state that the institution classifies as a violation state with  $i$  being the violator. Notice that Formula 13 formalizes at the same time an obligation pertaining to the role “buyer” and a right pertaining to the role “seller”. Of particular interest are then those consequences that attribute powers to agents enacting specific

roles:

$$rea(i, buyer) \sqcap rea(j, seller) \sqsubseteq \forall BID(i, j, b).bid(i, j, b) \quad (14)$$

$$SEND(i, j, msg49) \sqsubseteq BID(i, j, b) \quad (15)$$

If agent  $i$  enacts the buyer role and  $j$  the seller role, every time agent  $i$  bids  $b$  to  $j$  this action results in an institutional state testifying that the corresponding bid has been placed by  $i$  (Formula 14). Formula 15 states how the bidding action can be executed, which is, by sending a specific message to  $j$  ( $SEND(i, j, msg49)$ ).

Some observations are in order. As readers acquainted with deontic logic have probably already noticed, our treatment of the notion of obligation (Formula 13) makes again use of a standard reduction approach ([16]). More interesting is instead how the notion of institutional power is modeled. Essentially, the empowerment phenomenon is analyzed in term of two rules: one specifying the institutional effects of an institutional action (Formula 14), and one translating the institutional transition type in a brute one (Formula 15). Systems of rules of this type *empower* the agents enacting some relevant role by establishing a connection between the brute actions of the agents and some institutional effect. Whether the agents are actually able to execute the required ‘brute’ actions is a different issue, since agent  $i$  can be in some states (or even all states) unable to effectuate a  $SEND(i, j, msg49)$  transition. This is the case also in human societies: priests are empowered to give rise to marriages but if a priest is not in state of performing the required speech acts he is actually unable to marry anybody. There is a difference between “being entitled” to make a bid and “being in state of” making a bid ([4]). In other words, Formulae 14 and 15 express only that agents playing the buyer role are entitled to make bids. The actual possibility of performing the required ‘brute’ actions is not an institutional issue, but rather an issue concerning the implementation of an institution in a concrete system. We address this issue extensively in Section 3<sup>3</sup>.

**Institutive modules** We call institutive modules those sets of terminological axioms of an institution specification describing how states with certain institutional properties can be reached, for instance, how an agent  $i$  can reach a state in which it enacts role  $r$ . They can be seen as procedures that the institution define in order for the agents to bring institutional states of affairs about.

EXAMPLE 4. (An institutive module for roles) *The fact that an agent  $i$  enacts a role  $r$  ( $rea(i, r)$ ) is the effect of a corresponding enactment action  $ENACT(i, r)$  performed under certain circumstances (Formula 16), namely that the agent does not already enact the role, and that the agent satisfies given conditions ( $cond(i, r)$ ), which might for instance pertain the computational capabilities required for an agent to play the chosen role, or its capability to interact with some specific system’s infrastructures. Formula 17 specifies instead the procedure counting as an action of type  $ENACT(i, r)$ . Such a procedure is performed through a system infrastructure  $s$ , which notifies to  $i$  that it has been registered as enacting role  $r$  after sending the necessary piece of data  $d$  ( $SEND(i, j, d)$ ), e.g., a valid credit card number.*

$$\neg rea(i, r) \sqcap cond(i, r) \sqsubseteq ENACT(i, r).rea(i, r) \quad (16)$$

$$SEND(i, s, d) \sqsubseteq NOTIFY(s, i) \sqsubseteq REG(i, r) \quad (17)$$

<sup>3</sup>See in particular Example 6 and Definition 5

**Terminative modules** Analogously, we call terminative modules those sets of terminological axioms stating how a state with certain institutional properties can be left. Rules of this kind state for instance how an agent can stop enacting a certain role. They can be thus thought of as procedures that the institution defines in order for the agent to see to it that certain institutional states stop holding.

EXAMPLE 5. (A terminative module for roles) *Terminative modules for roles specify, for instance, how a transition type  $DEACT(i, r)$  can be executed which has as consequence the reaching of a state of type  $\neg rea(i, r)$ :*

$$rea(i, r) \sqsubseteq DEACT(i, r).\neg rea(i, r) \quad (18)$$

$$SEND(i, s, msg9) \sqsubseteq DEACT(i, r) \quad (19)$$

*That is to say,  $i$  deacting a role  $r$  always leads to a state where  $i$  does not enact role  $r$ ; and  $i$  sending message No.9 to a specific interface infrastructure  $s$  count as deacting role  $r$ .*

Examples 3-5 have shown how roles can be formalized in our framework thereby getting a formal semantics: roles are also sets of terminological axioms concerning state types of the sort  $rea(i, r)$ . It is worth noticing that this modeling option is aligned with work on social theory addressing the concept of role such as [20].

## 2.5 Tractable specifications of institutions

In the previous sections we fully deployed the expressivity of the language introduced in Section 2.1 and used its semantics to provide a formal understanding of many essential aspects of institutions in terms of transition systems. This section spends a few words about the viability of performing automated reasoning in the logic presented. The satisfiability problem<sup>4</sup> for logic  $\mathcal{ALCH}^{(\sqcup, \circ, \neg, id)}$  is undecidable since transition type inclusion axioms correspond to a version of what in Description Logic are known as “role-value maps” and logics extending  $\mathcal{ALC}$  with role-value maps are known to be undecidable ([3]).

Tractable (i.e., polynomial time decidable) fragments of  $\mathcal{ALCH}^{(\sqcup, \circ, \neg, id)}$  can however be isolated which still exhibit some key expressive features. One of them is logic  $\mathcal{ELH}^{(\circ)}$ . It is obtained from description logic  $\mathcal{EL}$ , which contains only state types intersection  $\sqcap$ , existential restriction  $\exists$  and  $\top$ <sup>5</sup>, but extended with the  $\perp$  state type and with transition type inclusion axioms of a complex form:  $a_1 \circ \dots \circ a_n \sqsubseteq a$  (with  $n$  finite number). Logic  $\mathcal{ELH}^{(\circ)}$  is also a fragment of the well investigated description logic  $\mathcal{EL}^{++}$  whose satisfiability problem has been shown in [2] to be decidable in polynomial time. Despite the very limited expressivity of this fragment, some rudimentary institutional specifications can still be successfully represented. Specifically, institutive and terminative modules can be represented which contain transition types inclusion axioms. Restricted versions of status

<sup>4</sup>This problem amounts to check whether a state description  $\gamma$  is satisfiable w.r.t. a given TBox  $T$ , i.e., to check if there exists a model  $m$  of  $T$  such that  $\emptyset \sqsubset \mathcal{I}(\gamma)$ . Notice that language  $\mathcal{ALCH}^{(\sqcup, \circ, \neg, id)}$  contains negation and intersection of arbitrary state types. It is well-known that if these operators are available then all most typical reasoning tasks at the TBox level can be reduced to the satisfiability problem.

<sup>5</sup>Notice therefore that  $\mathcal{EL}$  is a seriously restricted fragment of  $\mathcal{ALC}$  since it does not contain the negation operator for state types (operators  $\sqcup$  and  $\forall$  remain thus undefinable).

modules can also be represented enabling two essential deontic notions: “it is possible (respectively, impossible) to reach a violation state by performing a transition of a certain type”, and “it is possible (respectively, impossible) to reach a legal state by performing a transition of a certain type”. To this aim language  $\mathcal{L}_{ins}$  would need to be expanded with a set of state types  $\{\text{legal}(i)\}_{0 \leq i \leq n}$  whose intuitive meaning is to denote legal states as opposed to states of type  $\text{viol}(i)$ .

Fragments like  $\mathcal{ELH}^{(o)}$  could be used as target logics within theory approximation approaches ([24]) by aiming at compiling TBoxes expressed in  $\mathcal{ALCH}^{(u,o,-,id)}$  into approximations in those fragments.

### 3. FROM NORMS TO STRUCTURES

#### 3.1 Infrastructures

In discussing Example 3 we observed how “being entitled” to make a bid does not imply “being in state of” making a bid. In other words, an institution can empower agents by means of appropriate rules but this empowerment can remain dead letter. Similar observations apply also to deontic notions: agents might be allowed to perform certain transactions under some relevant conditions but they might be unable to do so under those same conditions. We refer to this kind of problems as *infrastructural*. The implementation of an institution in a concrete system calls therefore for the design of appropriate infrastructures or artifacts ([19]). The formal specification of an infrastructure amounts to the formal specification of interaction requirements, that is to say, the specification of which relevant transition types are executable and under what conditions.

**DEFINITION 4.** (*Infrastructures as TBoxes*) An infrastructure  $\mathfrak{Inf} = \langle \Gamma_{inf}, A_{inf} \rangle$  for institution  $\mathfrak{Ins}$  is a TBox on  $\mathcal{L}_{brute}$  such that for all  $a \in \mathcal{L}(A_{bridge})$  there exist terminological axioms in  $\Gamma_{inf}$  of the following form:  $\gamma \equiv \exists a. \top$  ( $a$  is executable exactly in  $\gamma$  states) and  $\gamma \equiv \exists \neg a. \top$  (the negation of  $a$  is executable exactly in  $\gamma$  states).

In other words, an infrastructure specification states all and only the conditions under which an atomic brute transition type and its negation are executable, which occur in the brute alphabet of the bridge axioms of  $\mathfrak{Ins}$ . It states what can be in concrete done and under what conditions.

**EXAMPLE 6.** (*Infrastructure specification*) Consider the institution specified in Example 1. A simple infrastructure  $\mathfrak{Inf}$  for that institution could contain for instance the following terminological axioms for any pair of different agents  $i, j$  and message type  $msg$ :

$$\top \sqsubseteq \exists \text{SEND}(msg33, i, j). \top \quad (20)$$

The formula states that it is always in the possibilities of agent  $i$  to send message No. 33 to agent  $j$ . It then follows on the grounds of Example 1 that agent  $i$  can always accept agent  $j$ .

$$\top \sqsubseteq \exists \text{ACCEPT}(i, j). \top \quad (21)$$

Notice that the executability condition is just  $\top$ .

We call a *concrete institution* specification  $\mathfrak{CIns}$  an institution specification  $\mathfrak{Ins}$  coupled with an infrastructure specification  $\mathfrak{Inf}$ .

**DEFINITION 5.** (*Concrete institution*)

A concrete institution obtained by the joining the institution  $\mathfrak{Ins} = \langle \Gamma_{ins}, A_{ins} \rangle$  and the infrastructure  $\mathfrak{Inf} = \langle \Gamma_{inf}, A_{inf} \rangle$  is a TBox  $\mathfrak{CIns} = \langle \Gamma, A \rangle$  such that  $\Gamma = \Gamma_{ins} \cup \Gamma_{inf}$  and  $A = A_{ins} \cup A_{inf}$ .

Obviously, different infrastructures can be devised for a same institution giving rise to different concrete institutions which makes precise implementation choices explicit. Of particular relevance are the implementation choices concerning abstract norms like the one represented in Formula 13. A designer can choose to regiment such norm ([15]), i.e., make violation states unreachable, via an appropriate infrastructure.

**EXAMPLE 7.** (*Regimentation via infrastructure specification*) Consider Example 3 and suppose the following translation rule to be also part of the institution:

$$\text{BNK}(i, j, b) \sqcup \text{CC}(i, j, b) \equiv \text{PAY}(i, j, b) \quad (22)$$

$$\text{condition\_pay}(i, j, b) \equiv \text{rea}(i, \text{buyer})$$

$$\sqcap \text{rea}(j, \text{seller}) \sqcap \text{win\_bid}(i, j, b) \quad (23)$$

The first formula states how the payment can be concretely carried out (via bank transfer or credit card) and the second just provides a concrete label grouping the institutional state types relevant for the norm. In order to specify a regimentation at the infrastructural level it is enough to state that:

$$\text{condition\_pay}(i, j, b) \equiv \exists (\text{BNK}(i, j, b) \sqcup \text{CC}(i, j, b)). \top \quad (24)$$

$$\neg \text{condition\_pay}(i, j, b) \equiv \exists \neg (\text{BNK}(i, j, b) \sqcup \text{CC}(i, j, b)). \top \quad (25)$$

In other words, in states of type  $\text{condition\_pay}(i, j, b)$  the only executable brute actions are  $\text{BANK}(i, j, b)$  or  $\text{CC}(i, j, b)$  and, therefore,  $\text{PAY}(i, j, b)$  would necessarily be executed. As a result, the following inclusion does not hold with respect to the corresponding concrete institution:  $\text{condition\_pay}(i, j, b) \sqsubseteq \exists \neg \text{PAY}(i) \text{viol}(i)$ .

#### 3.2 Organizational Structures

This section briefly summarizes and adapts the perspective and results on organizational structures presented in [14, 11]. We refer to that work for a more comprehensive exposition.

Organizational structures typically concern the way agents interact within organizations. These interactions can be depicted as the links of a graph defined on the set of roles of the organization. Such links are then to be labeled on the basis of the type of interaction they stand for. First of all, it should be clear whether a link denotes that a certain interaction between two roles *can*, or *ought to*, or *may* etc. take place. Secondly, links should be labeled according to the transition type  $\alpha$  they refer to and the conditions  $\gamma$  in which that transition can, ought to, may etc. take place. Links in a formal specification of an organizational structure stand therefore for statements of the kind: “role  $r$  can (ought to, may) execute  $\alpha$  w.r.t. role  $s$  if  $\gamma$  is the case”. For the sake of simplicity, the following definition will consider only the “can” and “ought-to” interaction modalities. State and transition types in  $\mathcal{L}_{ins} \cup \mathcal{L}_{brute}$  will be used to label the links of the structure. Interaction modalities can therefore be of an institutional kind or of a brute kind.

**DEFINITION 6.** (*Organizational structure*)

An organizational structure is a multi-graph:

$$\mathfrak{OS} = \langle Roles, \{C_p\}_{p \in Mod}, \{O_p\}_{p \in Mod} \rangle$$

where:

- *Mod* denotes a set of pairs  $p = \gamma : \alpha$ , that is, a set of state type (condition) and transition type (action) pairs of  $\mathcal{L}_{ins} \cup \mathcal{L}_{brute}$  with  $\alpha$  taken to be an atomic transition type;
- $\mathcal{C}$  (“can”) denotes links to be interpreted in terms of the executability of the related  $\alpha$  in  $\gamma$ , whereas  $\mathcal{O}$  (“ought”) denotes links to be interpreted in terms of the obligation to execute the related  $\alpha$  in  $\gamma$ .

By the expressions  $(r, s) \in \mathcal{C}_{\gamma:\alpha}$  and  $(r, s) \in \mathcal{O}_{\gamma:\alpha}$  we mean therefore: role  $r$  can respectively, ought to, interact with role  $s$  by performing  $\alpha$  in states of type  $\gamma$ .

As shown in [11] such formal representations of organizational structures are of use for investigating the structural properties (robustness, flexibility, etc.) that a given organization exhibits.

At this point all the formal means are put in place which allow us to formally represent institutions as well as organizational structures. The next and final step of the work consists in providing a formal relation between the two frameworks. This formal relation will make explicit how institutions are related to organizational structures and vice versa. In particular, it will become clear how a normative conception of the notion of role relates to a structural one, that is, how the view of roles as a sets of norms (specifying how an agent can enact and deact the role, and what social status it obtains by doing that) relates to the view of roles as positions within social structures.

### 3.3 Relating institutions to organizations

To translate a given concrete institution into a corresponding organizational structure we need a function  $t$  assigning pairs of roles to axioms. Let us denote with  $\text{Sub}$  the set of all state type inclusion statements  $\gamma_1 \sqsubseteq \gamma_2$  that can be expressed on  $\mathcal{L}_{ins} \cup \mathcal{L}_{brute}$ . Function  $t$  is a partial function  $\text{Sub} \dashrightarrow Roles \times Roles$  such that, for any  $x \in \text{Sub}$  if  $x = \text{rea}(i, r) \sqcap \text{rea}(j, s) \sqcap \gamma \sqsubseteq \exists \alpha. \top$  (executability) or  $x = \text{rea}(i, r) \sqcap \text{rea}(j, s) \sqcap \gamma \sqsubseteq \forall \neg \alpha. \text{viol}(i)$  (obligation) then  $t(x) = (r, s)$ . That is to say, executability and obligation laws containing the enactment configuration  $\text{rea}(i, r) \sqcap \text{rea}(j, s)$  as a premise are translated into role pairs  $(r, s)$ .

**DEFINITION 7.** (*Correspondence of specifications*)

A concrete institution  $\mathfrak{CIns} = \langle \Gamma, A \rangle$  is said to correspond to an organizational structure  $\mathfrak{OS}$  (and vice versa) if, for every  $x \in \Gamma$ :

- $x = \text{rea}(i, r) \sqcap \text{rea}(j, s) \sqcap \gamma \sqsubseteq \exists \alpha. \top$  iff  $t(x) \in \mathcal{C}_{\gamma:\alpha}$
- $x = \text{rea}(i, r) \sqcap \text{rea}(j, s) \sqcap \gamma \sqsubseteq \forall \neg \alpha. \text{viol}(i)$  iff  $t(x) \in \mathcal{O}_{\gamma:\alpha}$

Intuitively, function  $t$  takes axioms from  $\Gamma$  (i.e., the set of state type terminological axioms of  $\mathfrak{CIns}$ ) and yields pairs of roles. Definition 7 labels the yielded pairs accordingly to the syntactic form of the translated axioms. More precisely, axioms of the form  $\text{rea}(i, r) \sqcap \text{rea}(j, s) \sqcap \gamma \sqsubseteq \exists \alpha. \top$  (executability laws) are translated into the pair  $(r, s)$  belonging to the executability dimension (i.e.,  $\mathcal{C}$ ) of the organizational

structure w.r.t. the execution of  $\alpha$  under circumstances  $\gamma$ . Analogously, axioms of the form  $\text{rea}(i, r) \sqcap \text{rea}(j, s) \sqcap \gamma \sqsubseteq \forall \neg \alpha. \text{viol}(i)$  (obligation laws) are translated into the pair  $(r, s)$  belonging to the obligation dimension (i.e.,  $\mathcal{O}$ ) of the organizational structure w.r.t. the execution of  $\alpha$  under circumstances  $\gamma$ . Leaving technicalities aside, function  $t$  distills thus the terminological and infrastructural constraints of  $\mathfrak{CIns}$  into structural ones. The institutive, terminative and status modules of roles are translated into definitions of positions within a  $\mathfrak{OS}$ .

From a design perspective the interpretation of Definition 7 is twofold. On the one hand (from left to right), it can make explicit what the structural consequences are of a given institution supported by a given infrastructure. On the other hand (from right to left), it can make explicit what kind of institution is actually implemented by a given organizational structure. Let us see this in some more details.

Given a concrete institution  $\mathfrak{CIns}$ , Definition 7 allows a designer to be aware of the impact that specific terminological choices (in particular, the choice of certain bridge axioms) and infrastructural ones have at a structural level. Notice that Definition 7 supports the inference of links in a structure. By checking whether a given inclusion statement of the relevant syntactic form follows from  $\mathfrak{CIns}$  (i.e., the so-called subsumption problem of DL) it is possible, via  $t$ , to add new links to the corresponding organizational structure. This can be recursively done by just adding any new inferred inclusion  $x$  to the previous set of axioms  $\Gamma$ , thus obtaining an updated institutional specification containing  $\Gamma \cup \{x\}$ . This process can be thought of as the *inference of structural links from institutional specifications*. In other words, it is possible to use institution specifications as inference tools for structural specifications. For instance, the infrastructural choice formalized in Example 7 implies that for the pair of roles (buyer, seller), it is always the case that  $(\text{buyer}, \text{seller}) \in \mathcal{C}_{\top:\text{PAY}}$ . This link follows from links  $(\text{buyer}, \text{seller}) \in \mathcal{C}_{\top:\text{BNK}}$  and  $(\text{buyer}, \text{seller}) \in \mathcal{C}_{\top:\text{CC}}$  on the grounds of the bridge axioms of the institution (Formula 22).

Suppose now a designer to be interested in a system which, besides implementing an institution, also incorporates an organizational structure enjoying desirable structural properties such as flexibility, or robustness<sup>6</sup>. By relating structural links to state type inclusions it is therefore possible to check whether adding a link in  $\mathfrak{OS}$  results in a stronger institutional specification, that is, if the corresponding inclusion statement is not already implied by  $\mathfrak{Ins}$ . To draw a parallelism with what just said in the previous paragraph, this process can be thought of as the *inference of norms and infrastructural constraints from the specification of organizational structures*. To give a simple example consider again Example 6 but from a reversed perspective. Suppose a designer wants a fully connected graph in the dimension  $\mathcal{C}_{\top:\text{SEND}}$  of the organizational structure. Exploiting Definition 7, we would obtain a number of executability laws in the fashion of Formula 20 for all roles in *Roles* (thus  $2^{|Roles|}$  axioms).

Definition 7 establishes a correspondence between two essentially different perspectives on the design of open systems allowing for feedbacks between the two to be formally analyzed. One last observation is in order. While given a

<sup>6</sup>In [11] it is shown how these and analogous properties can be precisely measured within the type of structures presented in Definition 6.

concrete institution an organizational structure can be in principle fully specified (by checking for all -finitely many-relevant inclusion statements whether they are implied or not by the institution), it is not possible to obtain a full terminological specification from an organizational structure. This lies on the fact that in Definition 6 the strictly terminological information contained in the specification of an institution (eminently, the set of transition type axioms  $A$  and therefore the bridge axioms) is lost while moving to a structural description. This shows, in turn, that the added value of the specification of institution lies precisely in the terminological link they establish between institutional and brute, i.e., system level notions.

## 4. CONCLUSIONS

The paper aimed at providing a comprehensive formal analysis of the institutional metaphor and its relation to the organizational one. The predominant formal tool has been description logic. TBoxes has been used to represent the specifications of institutions (Definition 3) and their infrastructures (Definition 6), providing therefore a transition system semantics for a number of institutional notions (Examples 1-7). Multi-graphs has then been used to represent the specification of organizational structures (Definition 6). The last result presented concerned the definition of a formal correspondence between institution and organization specifications (Definition 7), which provides a formal way for switching between the two paradigms. All in all, these results deliver a way for relating abstract system specifications (i.e., institutions as sets of norms) to specifications that are closer to an implemented system (i.e., organizational structures).

## 5. REFERENCES

- [1] G. Azzoni. Il cavallo di caligola. In *Ontologia sociale potere deontico e regole costitutive*, pages 45–54. Quodlibet, Macerata, Italy, 2003.
- [2] F. Baader, S. Brandt, and C. Lutz. Pushing the  $\mathcal{EL}$  envelope. In *Proceedings of IJCAI'05*, Edinburgh, UK, 2005. Morgan-Kaufmann Publishers.
- [3] F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. Patel-Schneider. *The Description Logic Handbook*. Cambridge Univ. Press, Cambridge, 2002.
- [4] C. Castelfranchi. The micro-macro constitution of power. *ProtoSociology*, 18:208–268, 2003.
- [5] D. D. Harel and Kozen and J. Tiuryn. Dynamic logic. In D. Gabbay and F. Guenther, editors, *Handbook of Philosophical Logic: Volume II*, pages 497–604. Reidel, Dordrecht, 1984.
- [6] M. Esteva, D. de la Cruz, and C. Sierra. ISLANDER: an electronic institutions editor. In *Proceedings of AAMAS'02*, pages 1045–1052, New York, NY, USA, 2002. ACM Press.
- [7] M. Esteva, J. Rodríguez-Aguilar, B. Rosell, and J. Arcos. Ameli: An agent-based middleware for electronic institutions. In *Proceedings of AAMAS'04*, New York, US, July 2004.
- [8] M. S. Fox. An organizational view of distributed systems. *IEEE Trans. Syst. Man Cyber*, 11(1):70 – 80, 1981.
- [9] C. Ghidini and F. Giunchiglia. A semantics for abstraction. In R. de Mántaras and L. Saitta, editors, *Proceedings of ECAI'04*, pages 343–347, 2004.
- [10] D. Grossi, H. Aldewereld, J. Vázquez-Salceda, and F. Dignum. Ontological aspects of the implementation of norms in agent-based electronic institutions. *Computational & Mathematical Organization Theory*, 12(2-3):251–275, April 2006.
- [11] D. Grossi, F. Dignum, V. Dignum, M. Dastani, and L. Royakkers. Structural evaluation of agent organizations. In *Proceedings of AAMAS'06*, pages 1110 – 1112, Hakodate, Japan, May 2006. ACM Press.
- [12] D. Grossi, F. Dignum, and J.-J. C. Meyer. Context in categorization. In L. Serafini and P. Bouquet, editors, *Proceedings of CRR'05*, volume 136 of *CEUR Workshop Proceedings*, Paris, June 2005.
- [13] D. Grossi, J.-J. Meyer, and F. Dignum. Classificatory aspects of counts-as: An analysis in modal logic. *Journal of Logic and Computation*, October 2006. doi: 10.1093/logcom/exl027.
- [14] J. F. Hübner, J. S. Sichman, and O. Boissier. Moise+: Towards a structural functional and deontic model for mas organization. In *Proceedings of AAMAS'02*, Bologna, Italy, July 2002. ACM Press.
- [15] A. J. I. Jones and M. Sergot. On the characterization of law and computer systems: The normative systems perspective. *Deontic Logic in Computer Science*, pages 275–307, 1993.
- [16] J. Krabbendam and J.-J. C. Meyer. Contextual deontic logics. In P. McNamara and H. Prakken, editors, *Norms, Logics and Information Systems*, pages 347–362, Amsterdam, 2003. IOS Press.
- [17] J.-J. Meyer, F. de Boer, R. M. van Eijk, K. V. Hindriks, and W. van der Hoek. On programming karo agents. *Logic Journal of the IGPL*, 9(2), 2001.
- [18] D. C. North. *Institutions, Institutional Change and Economic Performance*. Cambridge University Press, Cambridge, 1990.
- [19] A. Omicini, A. Ricci, A. Viroli, C. Castelfranchi, and L. Tummolini. Coordination artifacts: Environment-based coordination for intelligent agents. In *Proceedings of AAMAS'04*, 2004.
- [20] I. Pörn. *Action theory and social science. Some formal models*. Reidel Publishing Company, Dordrecht, The Netherlands, 1977.
- [21] S. Pufendorf. *De Jure Naturae et Gentium*. Amsterdam, 1688. English translation, Clarendon, 1934.
- [22] A. S. Rao and M. P. Georgeff. Modeling rational agents within a BDI-architecture. In J. Allen, R. Fikes, and E. Sandewall, editors, *Proceedings of KR'91*, pages 473–484. Morgan Kaufmann: San Mateo, CA, USA, 1991.
- [23] D. W. P. Ruiter. A basic classification of legal institutions. *Ratio Juris*, 10:357–371, 1997.
- [24] M. Schaerf and M. Cadoli. Tractable reasoning via approximation. *Artificial Intelligence*, 74(2):249–310, 1995.
- [25] J. Searle. *The Construction of Social Reality*. Free Press, 1995.
- [26] J. Vázquez-Salceda. *The role of Norms and Electronic Institutions in Multi-Agent Systems*. Birkhuser Verlag AG, 2004.