A Distributed Normative Infrastructure for Situated Multi-Agent Organisations

Fabio Y. Okuyama[1], Rafael H. Bordini[2], and Antônio Carlos da Rocha Costa[3]

[1] Universidade Federal do Rio Grande do Sul, Brazil. okuyama@inf.ufrgs.br
[2] University of Durham, United Kingdom. R.Bordini@durham.ac.uk
[3] Universidade Católica de Pelotas, Brazil. rocha@atlas.ucpel.tche.br

Abstract. In most of the existing approaches to the design of multi-agent systems, there is no clear way in which to relate organisational and normative structures to the model of the environment where they are to be situated and operate. Our work addresses this problem by putting together, in a practical approach to developing multi-agent systems (and social simulations in particular), a high-level environment modelling language that incorporates aspects of agents, organisations, and normative structures. The paper explains in some detail how the ideas of *normative objects* and *normative places*, put together as a *distributed normative infrastructure*, allow the definition of certain kinds of *situated multi-agent organisations*, in particular organisations for multi-agent systems that operate within concrete environments. Normative objects are environment objects used to convey explicitly normative content that regulates the behaviour of agents within the place where such objects can be perceived by such agents. The paper defines these concepts, shows how they were integrated into the MAS-SOC multi-agent systems platform for social simulation, gives examples that illustrate the approach, and hints on new problems of (situated) organisational and normative structures that were brought forward by the work presented here.

1 Introduction

Multi-agent systems (MAS) are typically composed of agents, an environment, organisational structures, and means of interaction among those components. Organisational structures for multi-agent systems have been usually defined in a non-situated way, by which we mean independently of the environment where the system is to operate. In face of this issue, when a MAS is to be situated in a environment, there appears to be a 'gap' between the environment and the organisational structures, since no connection can be made between elements of the organisational structure and the physical places where such elements operate.

Furthermore, most current approaches to normative multi-agent systems [1] address the various issues on how norms can be defined, enforced, and so forth but with no clear indication on how those approaches can be used in the practical development of MAS. The work reported in this paper aims to address both these issues of state-of-the-art normative multi-agent systems.

At first sight, the connection between environment and organisation could appear to be unimportant for the modelling and understanding of the system. However, as one recognises that the physical environment may influence the operation of the organisation and of the agents that work in it, one also has to recognise that the explicit connection between organisational structures and environmental structures may be of some importance for the concrete realisation of such *situated organisations*.

In particular, the connection between organisational and environmental structures and processes is important when one is dealing with organisations whose normative structure does not operate in a homogeneous way throughout the physical environment. That is, it is important for organisations where the rules that regulate the behaviour of the agents vary according to the different places where the agents are located. For instance, in a certain factory, workers that are located in an excessively noisy room may be required to work no more than two consecutive hours with a break of at least twenty minutes, while workers that are located in a less noisy room may be allowed to work more consecutive hours without a break. Also, organisations often make use of norms that are spatially and temporally bounded — that is, norms that refer to some specific places and times, and not to others. Simple behavioural regulations are often of this kind, for example in signs such as "Please keep silence" and "Do not enter after 18:00h". Moreover, locations and physical objects may be used by organisations to the *empowering* of their agents. For example, porters are given the authority to check and control anyone entering or leaving a building, and one signal of that authority is the place where they conduct their work.

In other words, in many situations organisations regulate their operation by making use of physical resources (objects and places) as means for the propagation and instantiation of norms and agent powers. Lacking an explicit connection between organisational and environmental structures represents, thus, a conceptual gap between the modelling of organisations and their realisation in concrete multi-agent systems. Often the problem is not noticed because in most approaches norms are given at a very abstract level, which then has the effect of making it difficult for norms to be implemented in practice (i.e., to be formulated in a way that agents can interpret and follow them).

It is precisely the gap between environmental and organisational/normative structures that we intend to bridge in our current work and, importantly, in such a way that can be directly implemented through a combination of our previous work on environment modelling with an agent-oriented programming framework, as well as with existing organisational models and normative languages. In particular, we have extended the MAS-SOC [2] multi-agent based simulation platform to incorporate the approach presented in this paper. Based on [12], which presents the ELMS language for environment description, and on [14], where notions of a normative infrastructure is presented, we have developed an approach for relating environmental structures to normative organisational structures of multi-agent systems, as reported in this paper.

In brief, with the extensions proposed here, the ELMS language for environment description has been extended to support *situated organisations* through *situated norms* and *situated group structures*. This was done by two means: first, we developed a *distributed normative infrastructure*, which is the structure that allows the distribution of normative information over the spatial environment; and second, we defined a *normative principle*, associated with the design of MAS, and conceived as a special form of a conditional deontic rule, where an explicit condition on an agent's perception of a norm is included:

> Agent $\mathcal{A}$, when playing the relevant role to a norm $\mathcal{N}$ and being physically situated within the confines referred to by a normative object $\mathcal{O}$ carrying the norm $\mathcal{N}$, is expected to reason about following $\mathcal{N}$, if the agent perceived $\mathcal{O}$; otherwise, agent $\mathcal{A}$ is exempted from reasoning about $\mathcal{N}$.

In other words, the normative infrastructure is meant to provide the required elements to allow the agent to take the decision if it will follow the norms. Since our main goal is to provide means to enable the development of social simulations with cognitive agents, it is mostly the case where the norm-breaking behaviour would be desirable (to the designer of the simulation) in order to observe how the other agents (playing organisational roles) would react in such situation.

In this paper, we present the results we have achieved so far, and hint on some of the new avenues that the current work opened for further research. In Section 2, we give the background to our work, summarising the main ideas of the MAS-SOC platform, and the AgentSpeak language as interpreted by the ***Jason*** interpreter, which we use to program the agents' reasoning. In Section 3, we quickly review the environment language that we defined for describing physical environments. In Section 4, we summarise the idea of a distributed normative infrastructure for situated multi-agent organisations, introducing the concepts of *normative object* and *normative place*. In Section 5, we discuss the way norms are integrated into the MAS-SOC platform, in particular the way norms can be represented, contextualised, interpreted, and checked. Section 6 discusses various issues brought forward by our work. Section 7 discusses some related work. The conclusions of this paper are given in Section 8.

2 Background

We are developing a simulation platform called MAS-SOC (**M**ulti-**A**gent **S**imulations for the **SOC**ial Sciences). Its purpose is to provide a framework for the creation of agent-based simulations which do not require too much experience in programming from users, yet allowing users to use state-of-the-art agent technologies. In particular, it should allow for the implementation of simulations with *cognitive agents*.

In our approach, the agents' reasoning is specified in an extended version of AgentSpeak (an abstract programming language for BDI agents introduced by A. Rao [16]), as interpreted by ***Jason***, an agent platform based on Java [3].

The environment where agents are to be situated are specified in ELMS (**E**nvironment Description **L**anguage for **M**ulti-Agent **S**imulation), a language specially designed for the description of multi-agent environments. Further references about the ELMS language can be found in [13].

Until recently, the description of shared environments for situated multi-agent systems [20] had not been thoroughly addressed in the literature on agent-oriented software engineering, as the environment where agents of multi-agent systems were to be situated used to be considered as "given" rather than an essential part of the engineering of such systems.

In [14], extensions to ELMS were made in order to introduce a distributed normative infrastructure within a (simulated) physical environment. Through the use of such infrastructure, an organisation can be integrated into the environment. However, those extensions only provide the means to situate the organisations within an environment. The concepts of *normative objects*, *normative places*, and *norm supervisors* introduced to the language are meant to bridge the above mentioned gap between environment structures and organisation structures, helping to support the proper instantiation of an organisation in an environment.

3 The ELMS Language

According to [21], agents are computational systems situated in some environment, and are capable of autonomous action in this environment in order to meet their design objectives. Therefore, the environment has an important role in the specification, design and implementation of a MAS, whether it is the Internet, the real world, or some simulated environment. Even when the environment of the multi-agent system is the "real world" and the agent is embodied in a robot with sensors and effectors, the environment model should play a significant role in the design of the system. Any robot should have a set of sensors that give a predefined set of percepts that the robot will acquire when sensing the environment. Also, it should have a set of effectors that allow a restricted set of (parameterisable) actions. Thus, the possible sensor inputs and effectors output should be modelled first to facilitate the development of the software for the robot.

We understand by *environment modelling*, the modelling of all of the external world that an agent needs to know about for reasoning and deciding on courses of action. Agents themselves should also be considered components of the environment insofar as, from the point of view of an agent, any other agent is also part of the environment.

Thus, to define agents from this point of view, it is necessary to include in the description of the environment all properties that define the aspects of the "body" of each agent, which represent the properties of an agent that are perceptible to other agents. Furthermore, it is necessary to model explicitly the physical actions that agents are allowed to perform and the perception capabilities they have in their environment.

Through the definition of actions and perceptions, we define the *physical rules* of the environment (e.g., no one may see or pass through a wall) that must be satisfied in order to ensure the meaningfulness of the system model. This is in contrast to the norms of a organisational structure, which an agent may reason upon and decide to breach, with no implication to the meaningfulness of the system.

The objects that are part of an environment can be modelled as a set of properties and a set of reactions that characterise the behaviour of such objects in response to stimuli. They are treated as *reactive* components; only agents are pro-active, through their reasoning and deliberation processes.

From the point of view of the ELMS language, the deliberative activities of an agent are not relevant and are left out of the description of the environment, since they are internal to the agents (i.e., they are not physically perceived by the other agents). As mentioned before, in the MAS-SOC platform, the mental aspects of agents are described in the AgentSpeak language.

Below, we briefly review how an environment is described using this language. In section 3.1 we describe the modelling of agent "bodies", and in Section 3.2 the physical environment.

3.1 Modelling Agent Bodies

In the environment description, agents are characterised through the definition of classes of *agent bodies*, agent sensorial capabilities, and agent effective capacities.

Agent Body: the agent's characteristics that are perceptible to other agents. In our approach, classes of agent *bodies* are defined by a set of properties that characterise them and are perceptible to other agents. Each *body* is associated with a set of actions that it is allowed to perform and a set of environment properties that it can perceive.

Agent Sensorial Capabilities (Percepts): each sensorial capability is used to specify which environment properties will be perceptible to each agent that has a "body" with such capacity. It determines the environmental properties that will be sensed by the agent and the specific circumstances where they are possible. If the preconditions are all satisfied, then the values of those properties will be made available to the agent's reasoning procedure as the result of the agent's perception of the environment.

Agent Effective Capacities (Actions): these are the capabilities of performing actions in the environment that are made available to each class of agent body. Each definition of an action determines the environmental changes that such action can make when performed by an agent that has a body with such capacities. These changes are defined as assignments of values to the attributes of the environment. As the choice on (courses of) actions is meant to be part of the agent's "mind", something that is more naturally seen as a whole series of actions should not be implemented as one action available to agents at the environment level.

3.2 Environment Modelling

The environment is modelled by the definition of the objects available in the environment, the reactions that such objects might produce, and the spatial representation of the environment. Each of these form specific language constructs classified as follows.

Physical Environment Objects: the objects that are present in the environment. Agents interact with objects through the actions they perform in the environment, and by perceiving them. Object structures are defined by a set of properties that are relevant to the modelling and that could potentially be perceived by agents. The reactions that a class of objects can have is given by a list of identifying labels.

Object Reactions: the objects can "react", under specific circumstances, responding to actions performed by the agents in the environment.

Space Representation: the spatial representation of the environment can be done as a grid or a graph, according to the requirements of a particular simulation or design preferences. When a grid is used, the space is divided into cells forming a grid that represents the spatial structure of the environment, either in 2 or 3 dimensions. As for resources, each grid cell can have reactions associated with them. When a graph is used, the space is represented by *nodes* connected by *links*. As the cells of the grid, each node of the graph represents a spatial location, where reactions may occur in response to some action performed on it. The links represent connections between places that agents can follow, and may have weights or values associated with them, according to the needs of each particular project. In both grid- and graph-base spaces, the granularity of the spatial representation should follow the requirements of the application.

4 Normative Infrastructure

Certain real environments have objects aimed at informing "agents" about norms, give some advice, or warn about potential dangers. For example, a poster fixed on a wall in a library asking for *silence* is an object in the environment, but also informs about a norm that should be respected within that space. The existence of such signs, which we call *normative objects*, implies the existence of a regulating code in such context, which we call *situated norm*.

Situated norms are only meant to be followed within certain boundaries of space or time, and lose their effect completely if those space and time restrictions are not met. Another important advantage of modelling some norms as situated norms is the fact that the spatial and temporal context where the norm is to be followed is immediately determinable. Thus, the norm can be "pre-compiled" to its situated form, making it easier for the agents to operationalise the norm, and also facilitating the verification of norm compliance.

The *normative infrastructure* is intended to provide a means to inform the agents about the norms in a specific spatial context, allowing the agents to reason about norms of which they may have no previous knowledge. Also, the normative infrastructure has no specific enforcing nature, as the agents might not perceive (or may pretend to have not perceived) some of the normative objects, when they should.

In this section we present the extensions to ELMS that are meant to provide an infrastructure allowing the distribution of normative information within an environment. We refer as *normative infrastructure*, the concepts introduced in ELMS in order to allow the distribution of norms over the environment, while we use *normative structure* to designate a structured set of norms, some of which may be expressed by instances of normative objects and places, thereby regulating some agent organisation.

4.1 Normative Objects

Normative objects are "readable" by agents under specific individual conditions; that is, an agent can read a specific rule if it has the ability to perceive that type of object, at the location where the object is placed in the environment model. In the most typical case, the condition is simply being physically close to the object.

Such objects can be defined before the simulation starts, or can be created dynamically during the simulation. Each normative object can be placed in a collection of cells/nodes of the spatial representation of the environment. For example, a cell or group of cells of an environment grid can be used to represent a *normative place*, determining the first condition for the normative object being perceived: it is only within that normative place that the content of the normative object is relevant. The conditions under which the normative objects can be perceived are defined by the simulation designer using the constructs for defining perception conditions.

The normative information in a normative object is “read” by an agent through its usual sensing/perceptual abilities. It contains the norm itself and also meta-information, as follows:

Id: identification string, for the management of the normative object within the system.

Norm: a string that represents the normative information; this can be in any format that the targeted agents are be able to understand — for instance, AgentSpeak terms in the case of ELMS environments in the MAS-SOC platform. However, to enforce a uniform norm specification format over all applications, a fixed format should be adopted. For practical reasons, we have chosen the policy language REI [11] for such purpose.

Type: the type of the normative information contained in the object; it determines the level of importance (e.g., a warning, an obligation, a direction);

Issued by: agent or group that issued the norm;

Source: where the power underlying the norm issuance comes from; this could be the role that was being performed by the agent when issuing the norm, and the organisation (or group) that endorses such role;

Addressees: the organisational components (agent groups and agent roles) to which the normative information applies.

Placement: the set of normative places where the normative information applies. If omitted, the object is assumed to be valid everywhere in the environment, but normally only under the specific conditions determined by the designer (see the next item).

Condition: conditions under which the normative information can be perceived. The conditions can be associated with physical location, time, perception capabilities, spatial orientation of agents and objects, etc.

It is worth noting that norm-abiding behaviour is not related just to the existence of a normative object at some place. Beyond the existence of such object, it is necessary for the agent to physically perceive the normative object. Besides, autonomous agents will also reason about whether to follow or not the norm stated by the normative object, if and when they perceived it. This suggests that various specific problems should be tackled, concerning the study of agent reasoning about *situated normative objects*, a topic we discuss in Section 6.

4.2 Normative Places

Normative places are abstractions to define the boundaries of spatial locations where a set of related activities are done, or where groups of agents interact, and where some specific norms are valid and relevant. These places are also the physical spaces where the components of an organisational structure are located; that is, a *normative place* constitutes the spatial scope of an organisation, as well as the norms related to that organisation. The relevant normative information for each place is usually stored there, through the use of *normative objects*.

A normative place is defined simply by an identification label (a name) and the specification of its spatial boundaries, which is defined by the set of cells of the grid that are part of it (or, the nodes of the graph, according to the spatial representation being used). For each normative place, a set of *local roles* is defined to be located at such place, so that the roles that are present in such spatial context are regulated through norms embedded in the *normative objects* that are placed in that space.

A normative place may have intersections with other normative places, or may even be contained by another. For example, a “school” may be seen as a normative place encompassing a large portion of the environment grid where some of those grid cells refer to a normative place “classroom” and others to a normative place “library”.

The area covered by a normative place may increase or decrease during a simulation, since we are dealing with possibly dynamic environments, which may be associated with possibly dynamic organisations. Thus, the influence area of an organisation may expand or reduce dynamically, according to the requirements of the application, by changing the set of cells or nodes defined to belong to such normative place, which can be done by an agent empowered to effect such change. Such changes may occur under two circumstances: first, when the organisation deliberately rearranges the area where it needs to influence agent behaviour; and second, when the organisation acknowledges that the agent behaviour prescribed in a particular place has become more widely practiced by the agents themselves, so the organisation changes its area of operation (a normative structure) to reflect the actual (emergent) agent behaviour.

Similarly, different social behaviour might emerge if we rearrange the distribution of normative objects within a normative place where a particular organisation is situated, or if we create new normative objects. Clearly, these situations appear in many social situations, and having high-level abstractions available to model such situations can greatly facilitate the development of social simulations.

5 Using Norms

5.1 Norm Contextualisation

Normative objects are not supposed to be means of broadcasting general norms. The norms informed through normative objects should be *contextualised* (by the system designer or the agent that created the norm), incorporating specific information about the normative place where it is relevant.

As the spatial context of the norm is bounded and determined by its normative place, a generic abstract norm can be “pre-compiled” using such information, in order to make it less abstract. This process is meant to facilitate norm operationalisation, as such concrete norms are “ready to use” in the spatial scope where it is relevant. Other advantages of having less abstract norms are that the verification of norm compliance is facilitated and that they can reduce misinterpretations that could occur with abstract, non-contextualised norms.

For example, a norm that says “be kind to the elderly” can be quite hard to operationalise and verify, in general. However, in a fixed spatial context, such as a bus or a train, with the norm contextualised as “give up your seat for the elderly”, or in a street crossing with the norm contextualised as “help elderly people cross the street”, the norm would be much easier for the agents to interpret, and easier to verify using any norm-compliance checking mechanism.

5.2 Policy Language

In the MAS-SOC platform, the norms contained in the normative objects can be expressed simply as valid AgentSpeak predicates. In order to represent the norms in a uniform way in all simulations, we have adopted the policy language REI [11]. REI is a policy language aimed at the definition of policies for pervasive computing, being well-suited to our platform, as the pervasive computing paradigm is very close to the notion of having different normative places in a physical environment. In the REI language, there are constructs to define rights, prohibition, obligations, and dispensations. Also, the language has many other constructs, which include specific ones to solve conflicts, action specification, and rights delegation. A detailed description of the language can be found at `http://www.cs.umbc.edu/~lkagal1/rei`. Below, we present some of the main constructs available in the REI language, summarised from [11].

Rights, prohibitions, obligations, and dispensations:

`has(agent_ag, right(action_act, Conds)):` means that agent *agent_ag* has the right of executing action *action_act* if expression *Conds* is satisfied. The notion of *right* or *permission* can be interpreted as the deontic expression ∼O∼.

`has(agent_ag, prohibition(action_act, Conds)):` means that agent *agent_ag* is prohibited from executing action *action_act* if expression *Conds* is satisfied. The notion of prohibition can be interpreted as the deontic expression O∼.

`has(agent_ag, obligation(action_act, Conds)):` means that agent *agent_ag* is obliged to execute action *action_act* if expression *Conds* is satisfied. The notion of obligation can be interpreted as the deontic operator O.

`has(agent_ag, dispensation(action_act, Conds)):` means that agent *agent_ag* is dispensed from executing action *action_act* if expression *Conds* is satisfied. The notion of dispensation can be interpreted as the deontic expression ∼O.

Definition of priorities:

`overrides(A1,B1):` means that rule `A1` overrides rule `B1`.

5.3 Library of Norm-Considering Plans

In order to facilitate the programming of normative agents, we developed AgentSpeak plans to deal with the reasoning and deliberation about certain kinds of norms. Those plans are organised in files that can be imported from another AgentSpeak file by the use of the `include` directive in ***Jason***. Since such plans are available as plain AgentSpeak files, it is also possible to use them as templates to build customised plans according to the requirements of individual projects.

For example, in order to program an agent that never violates a prohibition to execute an action `a`, one should replace in its AgentSpeak program, every occurrence of `a` by `!execute(a)`, and also include the following plans in the agent's plan library:

```
+!execute(Action)
     : not prohibition(Action,_)
     <- Action.
```

```
+!execute(Action)
      : prohibition(Action,Condition)
        & not Condition
      <- Action.
+!execute(Action)
      : prohibition(Action,Condition)
        & Condition
    <- .fail.
```

As another example, to program an agent that always accomplishes an obligation determined by an organisation it trusts, unless the agent turns out to be dispensed from it (or, of course, if it violates some prohibition), the following plans can be used:

```
+has(Self,obligation(Action,Condition))[sourceOrg(SO)]
    :  .my_name(Self) & trusted(SO) & Condition
    <- !checkDispensation(Action);

+!checkDispensation(Action)
    :  .my_name(Self)
       & has(Self,dispensation(Action,Condition))
       & not Condition
    <- !execute(Action).

+!checkDispensation(Action).
```

In the code above, to handle the event that occurred because a new obligation was perceived, the agent checks whether the organisation that endorses the norm is trusted; if so, it checks the conditions of the obligation, and then it checks if there is a dispensation for such obligation; if there is none, it will execute the action, after checking if there is no prohibition on such action, as usual. The AgentSpeak code above is used to give priority to prohibition: a prohibited action is never executed. However, priorities among norms can be easily changed. In the code above, replacing `!execute(Action)` by `Action` would cause the agent to give priority on the obligation over the prohibition.

A topic of research in normative multi-agent systems is precisely the definition of general reasoning procedures to cope with various sorts of normative situations, such as when an action is both desired and prohibited [1]. We have not yet addressed some of these issues, in particular those that are controversial and still being debated at the theoretical level.

6 Issues in Distributing Norms

6.1 Norm Monitoring

In our approach, we define a special class of agents, called *norm supervisors*, which monitor other agents' compliance to norms within an organisation. Since agents are free to reason about abiding or not to a norm stated in a normative object, there is also the need to monitor the behaviour of those agents, at least in some applications. In order to be able to act as a norm supervisor, an agent may need extra information and perhaps extra capabilities. For this reason, it is possible to define, in ELMS, an agent as a *norm supervisor*, which will enable it to receive information about the relevant normative structure as well as about the actions being done by other agents in a given normative place.

Agents in charge of norm supervision could be agents that are not part of the actual simulation being conducted, designed specially to check agents' compliance to norms, or could be "normal" agents that belong to the simulation, and whose interests require that certain other agents follow certain norms. As the norm and the possible violations are confined to a specific normative place, it is potentially easier to identify the possible violations of those norms. The simulation designer may want to enable such capacity in a agent just to help it achieving its goal, to use such information to monitor/debug the simulation, or as an input to a reputation system, among other things.

For instance, according to [5], an agent may be motivated to verify the compliance to norms by other agents in order to reassure itself that the costs of norm adherence is being paid by the other agents too. A norm abiding agent will want that all the other addressees of the norms follow it too, otherwise the norm adhering behaviour may become some sort of competitive disadvantage. In [5], the authors refer to agents with such behaviour as "norm defenders".

6.2 Organisations and Environments

In most of the existing approaches for multi-agent organisations, such as Moise^{+} [10], the organisational structures are connected to the agent's reasoning by the implementation or through communication messages. Our work is not intended to replace such connection. In fact, we aim to intensify the interactions of agents and organisations, by having both direct an indirect interactions. The connection of an organisation to an environment, in our approach, can be done essentially in two ways:

Static: as shown in the left-hand side of Figure 1, from an (external) organisation description, the designer can model the normative structure to reflect a static image of an organisation, converting the organisational structures into roles and organisational links. The roles are attached to the normative places while the organisational behaviours and links are prescribed by norms included in normative objects.

Dynamic: as shown in the right-hand side of Figure 1, agent '*org ag*' obtains the information available in the organisation description and dynamically changes the normative structure in the environment. An agent, when receiving the percepts from the environment, may use this information as feedback to the organisational engine, which may change the organisation. Agent *'org ag'* may take part in the simulation or not, according to the requirements of the application.

It seems that, by using either static or dynamic binding as described above, it is possible to integrate an environment definition with most of the existing approaches to multi-agent systems organisations such as [18, 10, 8, 6]. However, simplifications may be required, and certain features of some approaches to agent organisations may not be captured by the use of such integration.

6.3 Implicit Role Adoption

For each normative place, a set of *local roles* that are regulated by the normative objects present in such place can be defined. In each spatially and temporally bounded normative place, an agent may adopt such temporary roles according to the activity it is doing in that place. The adoption of such roles may happen in an explicit or implicit manner.

An agent may have to explicitly identify itself to the other agents or institutions in order to adopt a specific role. For example, in a website that offers banking transactions,

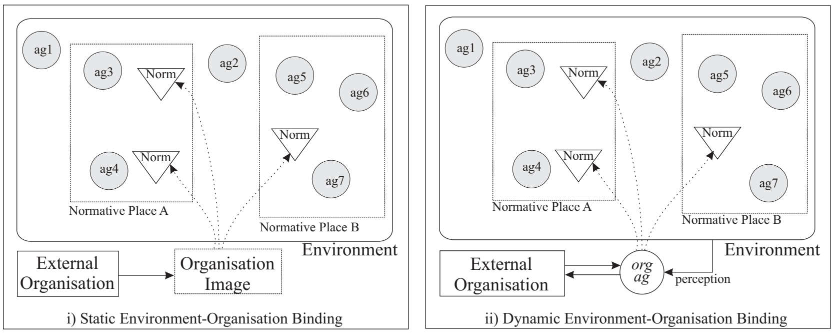

Fig. 1. Organisation and Environment Binding

an agent may browse anonymously the public sections, but if it wants access to private information about its account, it must authenticate itself (e.g., with a username and password), explicitly adopting the role of an "identified user", acquiring access to the specific rights of that role.

Using special elements (spatial positioning, orientation, possession of certain objects, agents' roles in organisational structures, etc.) an implicit role adoption may happen, which can be defined for each normative place with the use of simple rules. Below, we give some examples (using pseudocode) of how this can be done:

Default role: a default role may be defined for each place; for example, the *user* role, in a library:

```
default -> agent.role = user
```

Possession of an object: an agent may hold an object that associates it to a role; for example an agent carrying a badge may be assigned the role of *staff* in a library:

```
agent.hasBadge -> agent.role in library = staff
```

Positioning: according to its position, an agent may have a specific role; for example an agent in a moving car, seated at the driver's seat, will be assigned the role of *driver*:

```
agent in car AND agent.position = driver-position
 -> agent.role in car = driver
```

Relative role: a role in an organisation can be associated to the role performed in another organisation:

```
agent.role in university1 = researcher
AND agent at university2
 -> agent.role in university2 = visiting-researcher
```

6.4 Perception-Bounded Norm Reasoning

Given that a norm published through a normative object is only accessible as far as the normative object itself is accessible (i.e., perceptible), the normative reasoning concerning that norm is bounded by the boundaries of perception of the normative object. That is, when an agent did not follow a norm that was supposed to be followed at a given normative place, at least three different types of reason could explain that fact: (i) the agent really did not perceive the normative object; (ii) the agent perceived the object but not carefully enough to be able to grasp its normative content; and finally (iii) the agent correctly perceived the object and its normative content, but decided not to follow the norm.

The problem of norm abiding in normative situations, based on normative objects, then, has to take into account not only the possibility that agents autonomously decide to follow or not to follow the norms, but also the possibility that agents are not able to correctly perceive the normative objects. Issues such as responsibility, and others related to norm abiding, incorporate thus not only the usual aspects of rationality and affectivity, but also issues related to physical perception in concrete environments. Our approach, by bridging the gap between environment, organisations, and normative systems, has highlighted such issues.

6.5 Distribution of Normative Objects

Issues related to the adequacy of the distribution of normative objects in a normative environment also arise when dealing with the kind of distributed normative infrastructure that we are proposing. That is, guaranteeing that the set of normative objects is well distributed, and distributed in a satisfactory way, is an issue that should concern the agents responsible for issuing normative objects or the system designer.

Moreover, regulation of normative exceptions is also often made with normative objects (e.g., "It is forbidden to smoke in this building, except in the areas marked with a 'Smoking allowed' sign"). Thus, guaranteeing that the hierarchical structure of norms operating at a given normative place is well accessible and understandable to the agents is also an issue for the regulating agents.

6.6 Modular Design of Normative Places

Using the abstractions of *normative place* and *local roles*, the roles referred to in each normative place are strictly relative to the functions being performed in such place, which may have no direct relation to existing roles in other parts of the organisation. For example, consider a city as a multi-agent system. In a normative place "street", an agent driving a car will be playing the "driver" role, and another one not driving a car will be playing the "pedestrian" role — this is regardless of whether they are a school teacher or a hospital nurse in other places.

In such case, from the point of view of the design of such environment portion and normative structure, the agents are simply moving from one place to another, irrespective of the various other roles that they may be playing for other organisations within the city. The partitioning the environment into *normative places*, which can be done in a modular way, reducing the interdependence of each part, facilitates the modelling of large-scale simulations.

7 Related Work and Discussion

The notion of artifacts [19] and coordination artifacts [15] resembles, in some aspects, our notion of *normative objects*. However, they are clearly a different concept. Coordination artifacts are defined as runtime abstractions that encapsulate and provide a coordination service to the agents, but they express normative rules only implicitly, through their automatic effects on the actions of the agents. So their impositive impact on agent behaviour dismisses any need for normative reasoning on the part of the agents. In fact, coordination artifacts preempt the agents' freedom to overlook social norms and to decide not to follow them. In our work, rather than having a notion of objects that by their (physical) properties facilitate coordination, normative objects are used to store *symbolic* information that can be interpreted by agents, so that they can become aware of norms that should be followed within a well-defined location. Even if the general notion of artifact is similar to ELMS objects and could also contain symbolic normative information, one advantage of our approach is that it allows for a declarative language in which to represent the environment, which is executed by the interpreter to simulate the environment, whereas the existing implementations of artifacts provide a Java API with which to program the environment model.

Our choice in regards to normative objects has the advantage of keeping open the possibility of agent autonomy, as suggested in [4]. Agents are, in principle, able to decide whether to follow the norms or not when pursuing goals. Another important aspect is that normative objects are spatially distributed over a physical environment, with a spatial scope where they apply, and closely tied to the part of the organisation that is physically located in that space. Our work simplifies the way designers can guide the behaviour of each individual agent as they move around an environment where organisations are spatially located; this allows agents to adapt the way they behave in different social contexts.

The AGRE model, presented in [6], allows the definition of structures that represent the physical space, as "specialisations" of a generic space. However, we find that the social structures are not contextualised as they are in our work, leaving the social and physical structures rather unrelated.

Another important series of related work is that on Electronic Institutions [8]. The internal workings of an electronic institutions is given (in rough terms) as a state-machine where each state is called a "scene". Each scene specifies the set of roles that agents can perform in it, and a "conversation protocol" that the agents should follow when interacting within a scene. To traverse the series of scenes of the electronic institution, agents must do a sequence of actions in each scene, and also commit to certain actions in certain scenes, as the result of having performed certain other actions in certain other scenes. Our notion of normative place was inspired by such notion of scene, in giving it a physical, spatial reference where norms apply.

Similar to the electronic institutions approach, Computational Institutions [17] are defined as virtual organisations ruled by constitutive norms and regulative norms. In such institutions, organisational modelling uses coordination artifacts as building blocks, in a way that is very similar to our use of normative objects in spatially distributed organisations, but still keeping implicit in coordination artifacts the normative content imposed on the agents.

In [9], the notion of *governor* agents was introduced. Such agents aim to ensure that external agents fulfil all their social duties during the enactment of an electronic institution. The external agents interact with the environment through the governor agents, which also inform them about norms and possible actions. Governor agents differ from the *norm supervisors* presented here, as norm supervisors do not have as primary objective to avoid norm infringement, while governors aim to prevent external agents from performing actions that are not permitted in an institution.

In the research on normative multi-agent systems, various aspects of normative systems are discussed which are not directly dealt with in our model as yet, such as an explicit model of sanctions and norm enforcement, and a formal basis for norm representation and reasoning [1, 22, 7]. The advantage of our approach, however, is that is can be directly used in the development of practical multi-agent systems. We plan to incorporate increasingly sophisticated aspects of normative systems into our framework in the future.

8 Conclusions

We have presented an approach to integrate the modelling of environments and organisations, using a normative infrastructure that provides the means to distribute normative information over an environment. Such infrastructure, composed of *normative objects* and *normative places*, allows the spatial contextualisation of norms. The contextualisation of norms in a bounded spatial/temporal scope facilitates the operationalisation of the norms and the verification of compliance, and helps avoiding the misinterpretation of norms. Also, in our approach, a normative structure is a connection point relating environments and organisations, being a reflection of the organisation on its environment.

The distribution of norms over the environment, using normative objects, allows the environment to be partitioned in a modular way. Such partitioning facilitates an independent modelling of each part of the system, reducing the interdependence among the various parts, thus facilitating the modular modelling of the environment and organisations, taking advantage of the natural distribution of certain environments, with norms being associated only with the places where they should be followed, instead of requiring a central repository of norms. Therefore, the proposed normative infrastructure facilitates the design, development, and maintenance of large-scale multi-agent systems or simulations. Besides, this should pose less of a burden on agents that dynamically access normative information compared to approaches where all norms are made centrally available.

We believe that an explicit environment description is an important part of a multi-agent system, as thoroughly discussed in the literature on approaches to modelling multi-agent environments. Also, environment modelling facilitates the engineering of multi-agent systems as it is a stable point from where the agent reasoning and the organisational structures can be tuned to facilitate the development of agents and organisations. The notion of spatially distributed normative objects that we have introduced seems to serve well the integration of organisations and environments.

As future work, we plan not only to develop simulations using our platform, so as to further evaluate it and improve our approach, but also to study interesting issues related to the use of this approach. One issue to be investigated in future work is that having the norms spread over many independent spatial scopes may result in different reputations of a single agent over the environment, leading to a notion of *locality of reputation*. Another interesting aspect is that, being conditioned on the possibility of perceiving the existence of a normative object, the reasoning of agents that deal with normative objects is necessarily of a non-monotonic nature.

In summary, the normative reasoning required by the possibility of having normative places within the environment, each one with with its own organisational purposes and sets of norms, leads to many issues to be addressed in the future.

Acknowledgements

This work was partially supported by CNPq and FAPERGS.

References

1. G. Boella, L. van der Torre, and H. Verhagen. Introduction to normative multiagent systems. In G. Boella, L. van der Torre, and H. Verhagen, editors, *Normative Multi-agent Systems*, number 07122 in Dagstuhl Seminar Proceedings. Internationales Begegnungs- und Forschungszentrum fuer Informatik (IBFI), Schloss Dagstuhl, Germany, 2007. <http://drops.dagstuhl.de/opus/volltexte/2007/918> [date of citation: 2007-01-01].
2. R. H. Bordini, A. C. d. R. Costa, J. F. Hübner, A. F. Moreira, F. Y. Okuyama, and R. Vieira. MAS-SOC: a social simulation platform based on agent-oriented programming. *Journal of Artificial Societies and Social Simulation*, 8(3), 2005.
3. R. H. Bordini, J. F. Hübner, and M. Wooldridge. *Programming Multi-Agent Systems in AgentSpeak Using **Jason***. Wiley Series in Agent Technology. John Wiley & Sons, 2007.
4. C. Castelfranchi, F. Dignum, C. M. Jonker, and J. Treur. Deliberative normative agents: Principles and architecture. In N. R. Jennings and Y. Lespérance, editors, *Intelligent Agents VI, Agent Theories, Architectures, and Languages (ATAL), 6th International Workshop, ATAL '99, Orlando, Florida, USA, July 15-17, 1999, Proceedings*, volume 1757 of *Lecture Notes in Computer Science*, pages 364–378. Springer, 2000.
5. R. Conte and C. Castelfranchi. *Cognitive and Social Action*. UCL Press, London, 1995.
6. J. Ferber, F. Michel, and J.-A. Báez-Barranco. Agre: Integrating environments with organizations. In D. Weyns, H. V. D. Parunak, and F. Michel, editors, *Environments for Multi-Agent Systems, First International Workshop, E4MAS 2004, New York, NY, USA, July 19, 2004, Revised Selected Papers*, volume 3374 of *Lecture Notes in Computer Science*, pages 48–56. Springer, 2005.
7. N. Fornara and M. Colombetti. Specifying and enforcing norms in artificial institutions. In G. Boella, L. van der Torre, and H. Verhagen, editors, *Normative Multi-agent Systems*, number 07122 in Dagstuhl Seminar Proceedings. Internationales Begegnungs- und Forschungszentrum fuer Informatik (IBFI), Schloss Dagstuhl, Germany, 2007. <http://drops.dagstuhl.de/opus/volltexte/2007/909> [date of citation: 2007-01-01].
8. A. Garcia-Camino, P. Noriega, and J. A. Rodríguez-Aguilar. Implementing norms in electronic institutions. In F. Dignum, V. Dignum, S. Koenig, S. Kraus, M. P. Singh, and M. Wooldridge, editors, *AAMAS '05: Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, pages 667–673, New York, NY, USA, 2005. ACM Press.

9. A. García-Camino, J. A. Rodríguez-Aguilar, C. Sierra, and W. W. Vasconcelos. A distributed architecture for norm-aware agent societies. In M. Baldoni, U. Endriss, A. Omicini, and P. Torroni, editors, *Declarative Agent Languages and Technologies III, Third International Workshop, DALT 2005, Utrecht, The Netherlands, July 25, 2005, Selected and Revised Papers*, volume 3904 of *Lecture Notes in Computer Science*, pages 89–105. Springer, 2006.
10. J. F. Hübner, J. S. Sichman, and O. Boissier. $\mathcal{M}$OISE$^+$: Towards a structural, functional, and deontic model for MAS organization. In *AAMAS '02: Proceedings of the first international joint conference on Autonomous agents and multiagent systems*, New York, NY, USA, 2002. ACM Press.
11. L. Kagal, T. W. Finin, and A. Joshi. A policy language for a pervasive computing environment. In *4th IEEE International Workshop on Policies for Distributed Systems and Networks (POLICY 2003), 4-6 June 2003, Lake Como, Italy*, pages 63–74. IEEE Computer Society, 2003.
12. F. Y. Okuyama, R. H. Bordini, and A. C. da Rocha Costa. ELMS: An environment description language for multi-agent simulation. In D. Weyns, H. V. D. Parunak, and F. Michel, editors, *Environments for Multi-Agent Systems, First International Workshop, E4MAS 2004, New York, NY, USA, July 19, 2004, Revised Selected Papers*, volume 3374 of *Lecture Notes in Computer Science*, pages 91–108. Springer, 2005.
13. F. Y. Okuyama, R. H. Bordini, and A. C. da Rocha Costa. Spatially distributed normative infrastructure. In D. Weyns, H. V. D. Parunak, and F. Michel, editors, *Third International Workshop, E4MAS 2006, Hakodate, Japan, May 8, 2006, Selected Revised and Invited Papers*, volume 4389 of *Lecture Notes in Computer Science*, pages 203–220. Springer Berlin / Heidelberg, 2007.
14. F. Y. Okuyama, R. H. Bordini, and A. C. da Rocha Costa. Spatially distributed normative objects. In P. Noriega, J. Vázquez-Salceda, G. Boella, O. Boissier, V. Dignum, N. Fornara, and E. Matson, editors, *Proceedings of the International Workshop on Coordination, Organization, Institutions and Norms in Agent Systems II (COIN), held with ECAI, Riva del Garda, Italy, 28 August 2006.*, volume 4386 of *Lecture Notes in Computer Science*, 2007.
15. A. Omicini, A. Ricci, M. Viroli, C. Castelfranchi, and L. Tummolini. Coordination artifacts: Environment-based coordination for intelligent agents. In *AAMAS '04: Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 286–293, Washington, DC, USA, 2004. IEEE Computer Society.
16. A. S. Rao. Agentspeak(l): Bdi agents speak out in a logical computable language. In W. V. de Velde and J. W. Perram, editors, *Agents Breaking Away, 7th European Workshop on Modelling Autonomous Agents in a Multi-Agent World, Eindhoven, The Netherlands, January 22-25, 1996, Proceedings*, volume 1038 of *Lecture Notes in Computer Science*, pages 42–55. Springer, 1996.
17. R. Rubino, A. Omicini, and E. Denti. Computational institutions for modelling norm-regulated MAS: An approach based on coordination artifacts. In G. Lindemann, S. Ossowski, J. Padget, and J. Vazquez-Salceda, editors, *1st International Workshop "Agents, Norms and Institutions for Regulated Multi-Agent Systems" (ANI@REM 2005)*, AAMAS 2005, Utrecht, The Netherlands, 25 July 2005.
18. J. Vázquez-Salceda, V. Dignum, and F. Dignum. Organizing multiagent systems. *Autonomous Agents and Multi-Agent Systems*, 11(3):307–360, 2005.
19. M. Viroli, A. Omicini, and A. Ricci. Engineering MAS environment with artifacts. In D. Weyns, H. V. D. Parunak, and F. Michel, editors, *2nd International Workshop "Environments for Multi-Agent Systems" (E4MAS 2005)*, pages 62–77, AAMAS 2005, Utrecht, The Netherlands, 26 July 2005.
20. D. Weyns, H. V. D. Parunak, F. Michel, T. Holvoet, and J. Ferber. Environments for multiagent systems state-of-the-art and research challenges. In D. Weyns, H. V. D. Parunak,

and F. Michel, editors, *Environments for Multi-Agent Systems, First International Workshop, E4MAS 2004, New York, NY, USA, July 19, 2004, Revised Selected Papers*, volume 3374 of *Lecture Notes in Computer Science*, pages 1–47. Springer, 2005.

21. M. Wooldridge. Intelligent agents. In G. Weiß, editor, *Multiagent Systems—A Modern Approach to Distributed Artificial Intelligence*, chapter 1, pages 27–77. MIT Press, Cambridge, MA, 1999.
22. F. L. y Lopez, M. Luck, and M. d'Inverno. A normative framework for agent-based systems. In G. Boella, L. van der Torre, and H. Verhagen, editors, *Normative Multi-agent Systems*, number 07122 in Dagstuhl Seminar Proceedings. Internationales Begegnungs- und Forschungszentrum fuer Informatik (IBFI), Schloss Dagstuhl, Germany, 2007. <http://drops.dagstuhl.de/opus/volltexte/2007/933> [date of citation: 2007-01-01].