A Distributed Normative Infrastructure for Situated Multi-Agent Organisations*

(Short Paper)

Fabio Y. Okuyama
PPGC – UFRGS
CP 15064 – 90501-970
Porto Alegre, Brazil
okuyama@inf.ufrgs.br

Rafael H. Bordini
University of Durham
Dept. of Computer Science
Durham DH1 3LE, U.K.
R.Bordini@durham.ac.uk

A. C. da Rocha Costa
Univ. Católica de Pelotas
Escola de Informática/PPGINF
96.010-000 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 explicitly convey normative content that regulate the behaviour of agents within the place where such objects can be perceived by agents. The paper briefly introduces such concepts, showing how they were integrated into the MAS-SOC multi-agent systems platform for social simulation, and hints on new problems of (situated) organisational and normative structures that were brought forward by the work presented here.

Categories and Subject Descriptors

I.2.11 [**Computing Methodologies**]: Artificial Intelligence—*Multiagent systems*

General Terms

Design

Keywords

Situated Multi-Agent Systems, Situated Organisations, Normative Infrastructure, Environment Modelling, Normative Multi-Agent Systems.

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 made between elements of the organisational structure and the physical places where such elements operate. Furthermore, most of current approaches to normative MAS 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.

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 proper 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 certain importance for the concrete realisation of such *situated organisations*.

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 MAS.

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 brief, with the extensions proposed here, the ELMS environment description language supports *situated organisations* through *situated norms* and *situated group structures*. This is done by using two means: first, we developed a *distributed normative infrastructure*, which is the structure

*This work is partially supported by CNPq, CAPES, and FAPERGS.

Cite as: A Distributed Normative Infrastructure for Situated Multi-Agent Organisations (Short Paper), Fabio Y. Okuyama, Rafael H. Bordini and A. C. da Rocha Costa, *Proc. of 7th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2008)*, Padgham, Parkes, Müller and Parsons (eds.), May, 12-16., 2008, Estoril, Portugal, pp. XXX-XXX.

that allows the distribution of normative information over the spatial environment; and second, a *normative principle* associated to the design of the MAS, conceived as a special form of 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 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.

We are developing a simulation platform called MAS-SOC (**M**ulti-**A**gent **S**imulations for the **SOC**ial Sciences). In our approach, the agents' reasoning is specified in an extended version of AgentSpeak interpreted by ***Jason*** [1]. 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 [5].

We understand by *environment modelling* the modelling of all of the external world that an agent needs to know for reasoning and deciding on its course of action. Agents 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.

2. 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 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.

2.1 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. Each normative object can be placed in a collection of cells 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 in that normative place that the content of the normative object is relevant.

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, such as: an identification (*id*); a *type*, which stores the kind of the normative information contained in the object — it determines the level of importance (e.g., a warning, an obligation, a direction); the *issuer* which refers to the agent or group that issued the norm; the *source* which refers to 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 rule; and finally the *placement* defines 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.

Finally, the norm itself is represented by 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 for MAS-SOC simulations. However, to provide uniform norm specification across different applications, a common format should be adopted. One practical option would be to the use of the policy language REI [4] for such purpose.

2.2 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, such 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.

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 shrink dynamically, according to the requirements of the application, by the change of 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 in 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 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 simulations, and having high-level abstractions available to model such situations can greatly facilitate the development of such simulations.

3. USING NORMS

3.1 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 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 the misinterpretations of the 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 to the elderly", or in a street crossing with the norm contextualised as "help elderly people to cross the street", the norm would be much more easily interpreted by the agents, and more easily verified by any norm-compliance checking mechanism.

3.2 Library of Norm-Considering Plans

In order to facilitate the programming of the agents' reasoning, 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. 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 needs of individual designers.

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[1] 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, consider the following code:

```
+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).
```

[1]In ***Jason***, it is possible to use pre-processing to make all such substitutions automatic.

The code above can be used 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 code gives priority to prohibition: a prohibited action is never executed. However, priorities among norms could be changed easily.

4. ISSUES IN DISTRIBUTING NORMS

4.1 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. 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 at a given normative place.

As the norm and the possible violations are confined to a specific normative place, it is much 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 [2], 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.

4.2 Organisations and Environments

In most of the existing approaches for multi-agent organisations, 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. We actually aim to intensify the interactions of agents and organisations, by having both direct an indirect interactions. The binding of an organisation and an environment can be done essentially in two ways:

Static: as shown in the upper section 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 inserted into the normative places while the organisational behaviours and links are prescribed by norms attached to normative objects.

Dynamic: as shown in the lower section 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 a 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.

With these forms of environment–organisation integration, it seems possible to integrate an ELMS environment with most of the existing approaches to MAS organisations. Of course, simplifications may be required, and certain features of some approaches may not be captured by the use of such integration. However, our initial experience with the minimal model of MAS organisations proposed in [3] supports that assumption.

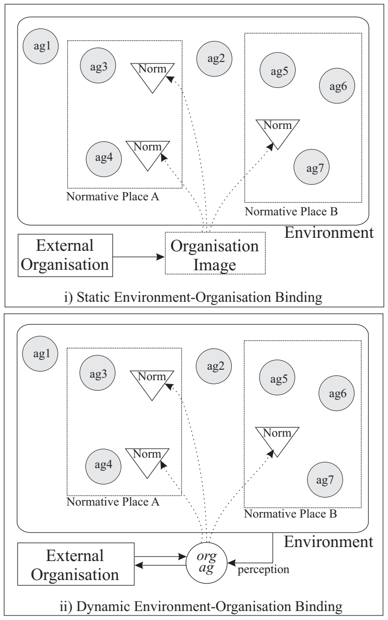

Figure 1: Organisation and environment binding.

4.3 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 environment, an agent may adopt such temporary roles according to the activity it is doing in such place. The adoption of such roles may happen in an explicit or implicit manner.

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. For example, in a library, while the default role of an agent would be the *user* role, by simply checking the possession of a specific badge by an agent, it could be assigned the role of *staff*. Another possibility is to verify the positioning of the agent: if an agent in a car is seated at the driver's seat, it would be assigned the role of *driver*. Also, the role played in one organisation may be used to assign the role in another one: while an agent plays the role of a *researcher* in a university, it plays the role of *visiting researcher* in another one.

4.4 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.

5. 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 by *normative objects* and *normative places*, allows the spatial contextualisation of norms. The contextualisation of norms in a bounded spatial 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 the environment.

The distribution of norms over the environment using normative objects allows the environment to be partitioned in a functional 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 organisation of each part, 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.

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 issue is the non-monotonic nature of normative reasoning on partially observable normative objects. More generally, the *norm reasoning* associated to the possibility of creating 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 future work.

6. REFERENCES

[1] R. Bordini, J. Hübner, and M. Wooldridge. *Programming Multi-Agent Systems in AgentSpeak Using* Jason. John Wiley & Sons, 2007.

[2] R. Conte and C. Castelfranchi. *Cognitive and Social Action*. UCL Press, London, 1995.

[3] Y. Demazeau and A. C. R. Costa. Populations and organizations in open multi-agent systems. In *1st Nat. Symposium on Parallel and Distributed AI (PDAI'96)*, India, 1996.

[4] L. Kagal, T. W. Finin, and A. Joshi. A policy language for a pervasive computing environment. In *4th IEEE Intl. Workshop on Policies for Distributed Systems and Networks, Italy*, p 63–74. IEEE, 2003.

[5] F.Okuyama, R. Bordini, and A. C. da Rocha Costa. Spatially distributed normative infrastructure. In *3rd Intl. Workshop E4MAS 2006, Japan*, v. 4389 of *LCNS*, Springer, 2007.