A Framework for Normative MultiAgent Organisations

Olivier Boissier and Jomi Fred Hübner*

SMA/G2I/ENSM.SE, 158 Cours Fauriel
42023 Saint-Etienne Cedex, France
FirstName.Name@emse.fr

Abstract. The social and organisational aspects of agency have led to a good amount of theoretical work in terms of formal models and theories. From these different works normative multiagent systems and multiagent organisations are particularily considered in this paper. Embodying such models and theories in the conception and engineering of proper infrastructures that achieve requirements of openness and adaptation, is still an open issue. In this direction, this paper presents and discusses a framework for normative multiagent organisations. Based on the Agents and Artifacts meta-model (A&A), it introduces organisational artifacts as first class entities to instrument the normative organisation for supporting agents activities within it.
Keywords: normative system, organisation, artifacts, norm enforcement

1 Introduction

These last years, the global landscape of multiagent technology has pointed out the concepts of norms and organisations for the modeling and programming of such systems [1] [26]. On one side, the introduction of norms have led to the notion of *normative multiagent system*. In [1], it is defined as "a multiagent system organized by means of mechanisms to represent, communicate, distribute, detect, create, modify, and enforce norms, and mechanisms to deliberate about norms and detect norm violation and fulfilment." On the other side, the increasing importance of organisations has promoted an *organisation oriented view* of the programming of MAS [2].

In this paper, we present a framework for *normative multiagent organisations*. Such an approach takes place at the intersection of the normative and the organisation approaches. In this framework, norms are anchored and considered in the context of the organisation of the system. Norms do not refer directly to agents but to primitives related to an organisation such as roles, groups, etc. The set of mechanisms cited above in the definition of normative systems, are

* Supported by French ANR Project ForTrust ANR-06-SETI-006.
[1] The series of COIN (Coordination Organisation Institution and Norms in agent systems) started in 2005 is an example of such an importance.

Dagstuhl Seminar Proceedings 09121
Normative Multi-Agent Systems
http://drops.dagstuhl.de/opus/volltexte/2009/1903

naturally enriched with functions related to the management of the organisation, to the support of the agents in their coordination and participation to the organisation.

As shown in [2], current software engineering approaches on the definition of these systems have led to a general architecture, kind of organisational middleware, composed of services and agents responsible for executing these mechanisms. From an architectural and software point of view, this middleware is generally introduced between the application agents and the agent communication platform. In those cases, the application agents do not have the possibility to take part in the management of the normative organisation to which they participate. As noticed in [20], the agents have become, in some sense, under the 'control' of the organisational middleware with respect to the management and use of *their* organisation. Our motivation in this work consists in the softening of the management of openness promoted by these organisational middlewares. We propose the implementation of the mechanisms supporting the normative organisation at the agents application level with first class entities.

The paper is structured as follows. The next section presents the main foundations that drive and structure our approach for the definition of the framework for normative multiagent organisation. In the following sections, we detail two components of this framework, starting by the *Organisation Modeling language* (cf. Sec. 3). The description of the different *Organisational Artifacts* that support the deployed normative multiagent organisation is splitted in those that are involved in the management and coordination of the organisation (cf. Sec. 4) and in those that support the management and regulation of the normative dimension of the organisation (cf. Sec. 5). The last component of the framework is composed of the *organisation-awareness mechanisms* that can be embedded in the agents of the systems to properly behave in such a framework. This latter component being still under development, some elements will be described instead of an exhaustive description. Before concluding, we provide some discussions and comparisons with the current state of the art. We position more particularily our approach with respect to different challenges presented in [1].

2 General view and foundations

In this section we present the foundational guidelines that have been used for the definition of the framework for the management of normative multiagent organisations. In the sequel, to alleviate the expression, we will use "organisation" instead of "normative organisation".

2.1 Different levels of representation of an organisation

A multiagent organisation can be considered and represented at three different levels: *(i)* the *organisation specification* stating the abstract structure and functioning of the MAS that is independent of the concrete agents that are participating to it, *(ii)* the *organisation entity* built by the different agents in

interaction within the organisation according to their autonomous interpretation and obedience of the specified organisation and *(iii)* the *internal organisation entities*, i.e. the local and individual representations of the organisation entity in every agent of the MAS. Let's notice that these levels are not independant. The organisation entity is updated and modified by runtime events related to agents entering and/or leaving the organisation, to group creation, to role adoption, to goal commitment, etc. The global representation of this organisation entity may be not accessible to the agents. It may be only represented in the eyes of an external observer. On the contrary a set of local, potentially inconsistent representations of it may be built and managed by each agent of the organisation. Agents may also be able to decide from these local representations, to adapt and to change the organisation in a bottom-up process, installing a new organisation specification.

To explicitly represent the organisation that is manipulated at these three levels, the framework is composed of an *Organisation Modeling Language* (OML). It is complemented by an *organisation implementation architecture* composed of the set of mechanisms to manage the organisation entity. This architecture is further divided into an organisation infrastructure part and into an agent part.

The *Organisation Modeling Language* (e.g. $\mathcal{M}$OISE$^+$ [24], ISLANDER [12]) is used to express the specification of the multiagent organisation in terms of norms, specific constraints and cooperation patterns that the designer (or the agents themselves) aim at imposing on the agents of the system. Several dimensions are considered in the current litterature: structural, functional, dialogic, etc [8]. One important feature of these OMLs is that norms and constraints do not refer directly to agents but to primitives related to an organisation such as roles, groups, etc. For instance, it can be specified that every agent that adopts a role "student" in a group "laboratory" is obliged to write a thesis. This language defines the explicit representation of the organisation at the three levels described above. Using these representations, the agents can reason on the organisation specification and on their local and individual representation of the organisation entity.

In the litterature, the development of the organisation implementation architecture normally considers both an agent-centred and an organisation-centred point of view [2]. The *agent-centered view* focuses on the organisational agent-level deliberative mechanisms to interpret and reason on the organisation specification and on the organisation entity to which the agents participate [4, 6]. Equiped with such *organisation-awareness mechanisms* agents become *organisation-aware agents*. Let's note that in the sequel, when we will use the term "agent" it is implicitly considered that the agent is an organisation-aware agent. The *organisation-centered* view is mainly concerned with the definition of what we call *organisational infrastructure* (OI) to support, interpret and manage the organisation entity derived from the enactment by the agents of the organisation

[2] In [35] these points of view are called agent and institutional perspectives.

specified with the OML. Thus, the OI provides the agents with global and shared mechanisms related to their participation to the organisation entity.

2.2 Regimented Norms vs Enforced Norms

As stated in the Sec. 1 and [10], a normative multiagent organisation serves as an instrument to control the autonomy of the agents. Its success depends on how the behavioural constraints stated in its specification are ensured in the system. These behavioural constraints are established by the norms that are stated by the specification of the organisation. In the context of this work, a norm can be an obligation, a permission, or an interdiction to perform some action or to achieve some goal. The actions and goals that are considered are related to the problem to solve (e.g. changing the state of some resource) but also related to the management of the organisation itself (e.g. adopting a role, entering in a group). A norm also has a condition that states when it is active and a deadline to be fulfilled [3]. These norms are considered and interpreted in the context of the current organisation entity. Two types of mechanisms can be considered for instrumenting them in the organisation [4]: *regimentation* and *enforcement*.

Regimentation is a mechanism that simply prevents the agents to perform actions that are stated as forbidden by a norm. More precisely, some actions are regimented in order to preserve important features (e.g. wellformedness) of the organisation. For instance, if a group can have at most one agent playing a given role, the organisational action 'adopt this role' in this group is regimented in order to ensure that this constraint is strictly respected. Since this mechanism has to work in an open system, for any kind of agents, it is implemented 'outside' the agents in the organisation infrastructure. Therefore, action regimentation implies the requirement to instrument the MAS with mechanisms preventing the execution of the concerned set of actions and to install them under the strict control of the OI.

Enforcement is a mechanism which is applied after the detection of the violation of some norm. While regimentation is a preventive mechanism, enforcement is a reactive one. From the local point of view of an agent, a norm may be decided to be obeyed or not. From the global point of view of the organisation, the fulfilment/unfulfilment of the norms should be detected, evaluated as a violation or not, and then judged as worth of sanction/reward or not. While detection can be implemented as an automatic process that does not require decision, the evaluation and the judgement need deliberation and reasoning.

[3] We are aware that the concept of norm is broader and more complex than the one used in this paper (e.g. [34] and the Deontic Logic in Computer Science workshop series [18]). For the present paper however this simple and informal definition is enough to discuss the proposal.

[4] This classification is based on the proposal described in [19, 15]. However, we present them in a more specific context: regimentation is applied only to preventing the execution of organisational actions and enforcement is applied for the the other cases.

Instrumenting norms of the organisation either as regimentations or enforcement mechanisms depends on which side the designer wants to give more weight. Looking further at the functions used in the corresponding mechanisms, two classes can be indeed distinguished: *(i)* management of regimentation in terms of interpretation of the considered norms and checking their satisfaction before executing changes in the organisation entity, *(ii)* management of enforcement in terms of status detection, evaluation of this status and judgement on violation or not followed by sanction execution.

2.3 First class entities for the Management of Normative Organisation

As mentioned in the introduction, the engineering of organisation infrastructure in the litterature has led to the proposals of organisation middleware installing the OI as a separate layer that cannot be managed by the agents participating to the application which is developed on top of this middleware. However, as argued in [20], even if the OI aims at supporting and controling the agents in their participation to the organisation entity, we consider that it should also be managed by those agents.
To solve this problem, we propose to design and develop it *within* the multiagent layer where the application is developed with the first class abstractions that are used to develop it. The choice of these first class abstractions must be considered with care, since, as stated in the previous section, the management of norms in the OI strongly depends on a regimentation or an enforcement view.
Basing our approach on the basic A&A (Agents and Artifacts) meta-model presented in [33], the organisation infrastructure of the framework, called ORA4MAS [25], proposes a set of artifacts, called *organisational artifacts*. The agents can use these artifacts to instrument *(i)* the multiagent environment which is no more a merely passive source of agent perceptions and target of agent actions and *(ii)* the organisation entities living upon in order to interpret and manage them according to the way they are specified with the OML. Given the deliberative nature of some of the mechanisms involved in the norm enforcement (evaluation, judgement and sanction), the overall picture of ORA4MAS accounts also for *organisational agents* (cf. Fig. 1).
We use here the adjective "organisational" to identify those agents and artifacts of the MAS which are part of the OI. They are responsible for activities and encapsulate functionalities concerning the management and enactment of the organisation. It is however possible, depending on the application requirements, that agents participating also to the proper solving and functioning of the application endorse the "role" of organisational agents.

Analogously to the human case, *organisational artifacts* are used here to reify and modularise the functional-part of the organisation management machinery. As in the A&A model, they are non-autonomous function-oriented entities, designed to provide resources and tools that agents can create and use. They are focused on the organisation entity management activities. As the cognitive

artifacts proposed in the A&A model, they constitute a distributed set of organisational resources and tools that can be perceived and used by agents as first-class entities. They can be dynamically adapted and possibly replaced (by agents themselves) during the organisation lifetime. As cognitive artifacts, the organisational artifact function is partitioned in a set of operations, which agents can trigger by acting on artifact usage interface. The usage interface provides all the controls that make it possible for an agent to interact with an organisational artifact, that is to use and observe it. Agents can use an organisational artifact by triggering the execution of operations through the usage interface and by perceiving observable events generated by the artifact itself, as a result of operation execution and evolution of its state. Besides the controls for triggering the execution of operation, an organisational artifact can have some observable properties, i.e. properties whose value is made observable to agents, without necessarily executing operations on it. Organsational artifacts then mediate the access of agents to organisation resources and support participation of these agents to organisation activities. For instance, to adopt a role an agent has to use the appropriate artifact.
Considering the normative dimension of the organisation, organisational artifacts encapsulate also organisational norms and functionalities, such as enabling, mediating, and ruling agent interaction, tracing and ruling resource access, and so on. Regimentations of norms (see Fig. 1) are implemented in the organisational artifacts. For instance, let's consider the case of a regimented adoption of role. The operation will be successfully executed only in the case the agent is allowed to adopt the role, otherwise the adoption fails. Since it is possible to link organisational artifacts with cognitive artifacts that mediate the access of agents to resources, it is thus possible to imagine to regiment also the access to those resources by the way of the organisational artifact. In the case of enforcement of norms, the functionality provided by the artifacts consists in the *detection* and showing (by means of observable properties) the non fulfilment of a norm. We consider that agents (organisational ones or not depending on the application) should be informed of current status of the norm and can evaluate the existence of violation or not and take the better decision regarding the application objectives.

The *organisational agents* embed dedicated reasoning and strategies related to the management of the organisation. They can be dedicated agents or agents of the application having special knowledge. They dynamically articulate, manage, regulate and adapt the organisation entity by creating, linking and manipulating the organisational artifacts, which are discovered and used by the agents to work inside the organisation entity, according to the specified organisations. Such activities typically include observing artifacts dynamics and possibly intervening, by changing and adapting artifacts or interacting directly with other agents, so as to improve the overall (or specific) organisation processes or taking some kinds of decisions when detecting violations. As an example, in the context of the $\mathcal{M}$OISE$^+$ model, one or multiple *scheme manager* agents can be introduced, responsible for monitoring the dynamics of the execution of a scheme by ob-

serving a specific artifact. The scheme artifact and scheme manager agents are designed so as that the artifact allows for violation of the deontic rules concerning the commitment of missions by agents playing some specific roles, and then the decision about what action to take – after detecting the violation – can be in charge of the manager agent.

Organisational artifacts and organisational agents create a sort of explicit organisational infrastructure on which the organisation entity is deployed, revealed to the agents as available tools in the environment. Regimentation and detection mechanisms into artifacts, whereas evaluation and judgement mechanisms involved in inforcement are implemented into the agents. ORA4MAS is thus able to ensure that important properties of the organisation entity hold while agents keep their autonomy with respect to the constraints considered as norms to enforce.

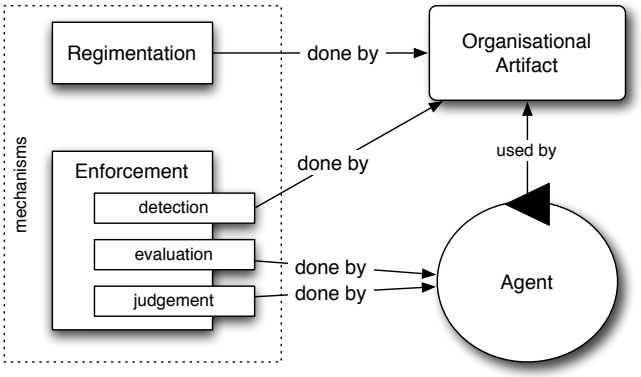

Fig. 1. General relation between the mechanisms that implement the norms and the organisational agents and artifacts of ORA4MAS

Given the sketching of the foundational guidelines underlying our framework, we describe in the next section its different components and how the full-fledged $\mathcal{M}$OISE$^+$ organisational model [5] can be implemented with organisational artifacts. Detailed examples of the use of the framework can be found in [20] and in [22]. ORA4MAS is realised on top of CARTAGO infrastructure [32], embedding algorithms used in $\mathcal{S}$-$\mathcal{M}$OISE$^+$ [23]. CARTAGO is integrated with ***Jason*** [3], 2APL [9], and JADEX [29] — these integrations are presented in [31, 28].

In this paper, descriptions of the organisation-awareness mechanisms and of the organisational agents are not given. Even if some work has been realized in the different examples that we have developed, we don't have yet generic

[5] Different OMLs require a different set of suitable artifacts and agents. For instance, in the AGR organisational model [14], we can conceive artifacts to manage groups; for ISLANDER [12], the artifacts can be used to manage the scenes.

architectures of such agents. Examples of organisation-awereness mechanisms may be found for instance in [5].

3 Normative Organisation Modeling Language

The current version of the framework uses the $\mathcal{M}$OISE$^+$ OML[24] as the language to define and describe explicitly an organisation. This language decomposes the specification of an organisation into three independent dimensions: structural, functional, and deontic dimensions [6]. The structural dimension focuses on the specification of the *roles*, *groups*, and *links* of the organisation. The definition of roles states that when an agent decides to play some role in a group, it is accepting some behavioural constraints related to this role. The functional dimension specifies how the *global collective goals* should be achieved, i.e. how these goals are decomposed into global *plans*, grouped into coherent sets (called *missions*) to be allocated to roles. The decomposition of global goals results in a goal-tree, called *scheme*, where the leaves-goals can be achieved individually by agents. The deontic dimension glues the structural dimension with the functional one by the specification of the roles' *permissions* and *obligations* for missions.

The detailed syntax and definition of the language is described in [21]. Let's stress that, agents and the OI interpret that declarative organisation specification. This language is founded on components represented by predicates and functions. Considering an organisation specification, $\mathcal{G}$, $\mathcal{R}$, $\mathcal{S}$, $\mathcal{M}$, Φ denote respectively the set of all group specifications, the set of all roles, the set of all scheme specifications, the set of all missions, and the set of all goals. We present here only some predicates that are used in the sequel of the paper:

- $compat(g, \rho, C)$: is true iff the role ρ ($\rho \in \mathcal{R}$) is *compatible* with all roles in the set C ($C \subseteq \mathcal{R}$) when played in the group g ($g \in \mathcal{G}$) (two roles are compatible if they can be adopted by the same agent);
- $mission\_scheme(m, s)$: is true iff the mission m ($m \in \mathcal{M}$) *belongs to* the scheme s ($s \in \mathcal{S}$);
- $goal\_mission(\varphi, m)$: is true iff the goal φ ($\varphi \in \Phi$) *belongs to* the mission m ($m \in \mathcal{M}$);
- $obl(\rho, m)$: is true iff the role ρ has an *obligation relation* to the mission m;
- $per(\rho, m)$: is true iff the role ρ has a *permission relation* to the mission m;
- $goal\_role(\varphi, \rho)$: is true iff goal φ is part of one of the obliged missions of role ρ;

Similarly, functions of this language that are considered in the sequel are:

[6] Extensions are currently on the way to integrate an extended version of it in the framework. These extensions have been developped in the Moise-Inst OML [17]. This OML proposes an enriched deontic dimension with more expressive normative expressions and also a supplementary dimension, called context specification, stating the a priori evolution of the organisation.

- $maxrp : \mathcal{R} \times \mathcal{G} \rightarrow \mathbb{Z}$: returns the maximum number of players of a role in a group, i.e. upper bound of the *role cardinality*;
- $minrp : \mathcal{R} \times \mathcal{G} \rightarrow \mathbb{Z}$: returns the minimum number of players of a role within a group, necessary for that group to be considered *well-formed* (i.e. lower bound of the *role cardinality*);
- $maxmp : \mathcal{M} \times \mathcal{S} \rightarrow \mathbb{Z}$: returns the maximum number of agents that can commit to a mission in a scheme (i.e. upper bound of the *mission cardinality*);
- $minmp : \mathcal{M} \times \mathcal{S} \rightarrow \mathbb{Z}$: returns the minimum number of agents that have to commit to a mission within a scheme for that scheme to be considered well-formed regarding that mission (i.e. lower bound of the *mission cardinality*).

4 Organisational Artifacts for Organisation Coordination

Derived from the OML presented in previous section, we describe here and in Sec. 5 the basic set of artifacts of ORA4MAS [25] that constitutes the building blocks for the support of the 'reification' of the structural specification (SS), functional specification (FS), and deontic specification (DS) of $\mathcal{M}$OISE$^+$. We focus here on the management of the organisation. In Sec. 5, we will present organisational artifacts in relation to the normative content of the organisation entity.

The basic set of organisational artifacts considered here accounts for: OrgBoard, GroupBoard artifacts, SchemeBoard artifacts. The instrumentation of the organisational entity with those artifacts is done as follows: one and only one OrgBoard is used to keep track of the current state of the deployed organisational artifacts supporting the current organisational entity in the overall, one GroupBoard for each instance of group of agents used to manage the life-cycle of this specific instance of a group, one SchemeBoard for each social scheme being executed by the agents used to support and manage the execution of it.

The organisational artifacts are linked together to allow the synchronisation of some their operations and to share information required to maintain a coherent and consistent state of the organisation entity. The OrgBoard is linked to all the other organisational artifacts of the organisation entity. The GroupBoard is linked to all SchemeBoard that manage schemes involving for their execution agents that are member of the corresponding group. Each SchemeBoard is linked to exactly one NormativeBoard (see next section) that verifies the status of the norms related to the execution of the scheme.

In the following we briefly describe these artifacts. We consider just a core set of the characteristics and functioning of the artifacts, skipping most details that would make heavy the overall understanding of the approach.

4.1 OrgBoard artifact

An abstract representation of the OrgBoard is depicted in Fig. 2. The observable properties of this organisational artifact are:

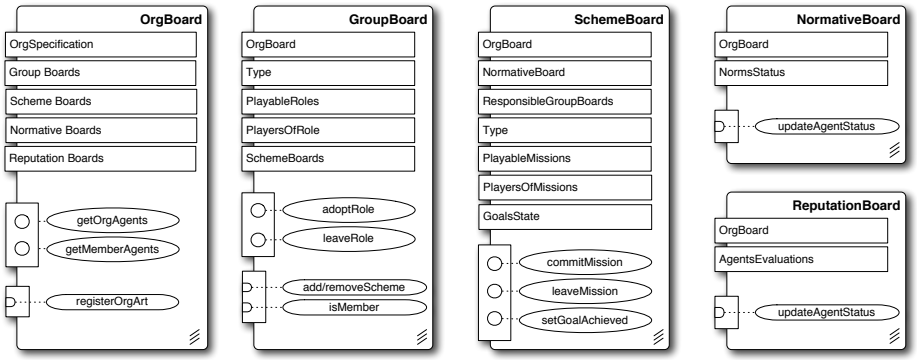

Fig. 2. Basic kinds of artifacts in ORA4MAS, with their usage interface, including operations (represented by circles) and observable properties (represented by rectangles with circles in their center), and the link interface (represented by rectangles with circles on their left side)

- OrgSpecification: specification of the organisation entity written in the $\mathcal{M}$OISE$^+$ OML. Agents may use this observable property to get the organisational specification. They can then reason about it and decide whether they want or no to enter in the organisation.
- GroupBoards, SchemeBoards, and NormativeBoards and ReputationBoards: identifiers of all instances of GroupBoard, SchemeBoard, NormativeBoard, ReputationBoard, respectively, within the organisation entity. Generally speaking, these observable properties make it possible for agents observing an OrgBoard to know the current set of organisational artifacts instrumenting the organisation entity.

The usage interface of the OrgBoard has the following operations:

- getOrgAgents(): used to get the set of agents having the status of organisational agent in the organisation entity (this status is set during the deployment of the system).
- getMemberAgents(): used to get the set of all agents playing at least one role within a group of the organisation entity [7]

The main link operation of the OrgBoard is registerOrgArt. It is used in the initialisation process of each new instance of GroupBoard, SchemeBoard, and NormativeBoard to be registered as an artifact linked to the organisation entity represented by the OrgBoard.

[7] This operation would be implemented as an observable properties; however due to distributed characteristic of the information (they are managed by all GroupBoard), maintaining an uptodate observable property would be time consuming. Using an operation, the list of member agents is a cache, computed only on demand.

4.2 GroupBoard artifact

A GroupBoard is an organisational artifact providing functionalities to manage a group. Each GroupBoard is attached to a group in the organisation entity instanciated from a specific group specification. It maintains a consistent state of that group by regimenting some norms stating essential structural properties. For instance, whenever some agent asks for a role adoption in the group managed by the GroupBoard, the GroupBoard regiments a set of norms that state when a role can be adopted: (1) the role belongs to its group specification; (2) each role that the agent already plays is specified as compatible with the new role; and (3) the number of players is lesser or equals than the maximum number of players defined in the group's compositional specification.

As an artifact, the GroupBoard has some observable properties (Fig. 2) that enable agents to know which are the available roles and their constraints, which are the participant agents, and which are the other organisational artifacts linked to the GroupBoard. Among them, the most relevant are:

- OrgBoard: is the reference to the OrgBoard that represents the organisation entity to which the group belongs.
- Type: is the identification of the group specification in the structural specification (an element of $\mathcal{G}$).
- PlayableRoles: contains all roles that can still be adopted in this group, i.e. those which the number of players is not the maximum yet. This property changes whenever a new agent enters into the group by adopting a role.
- PlayersOfRole: contains the names of all agents belonging to the group and their corresponding roles.
- SchemeBoards: contains a set of all schemes the group is responsible for.

The usage interface accounts for the following operations:

- adoptRole(ρ): used by an agent to adopt a new role in the group, where $\rho \in \mathcal{R}$ is the identifier for a role in the Structural Specification.
- leaveRole(ρ): used by an agent to give up the role ρ that it had adopted previously.

The link operations of a GroupBoard manage the coordinations with its linked organisational artifacts. Among them, we have:

- addSchemeBoard(sb): used by a SchemeBoard initialisation process to notify the GroupBoard that it is responsible for the scheme sb. The GroupBoard updates accordingly its SchemeBoards observable property.
- removeSchemeBoard(sb): used by a SchemeBoard linked to the GroupBoard, to notify that the GroupBoard is no more responsible for the execution of the scheme sb. The GroupBoard updates accordingly its SchemeBoards observable property.
- isMember(α): used by a SchemeBoard to request whether an agent α is member (i.e. is playing at least one role) of the group managed by the GroupBoard.

Before presenting the norms that a GroupBoard regiments, let's introduce some predicates related to the internal state of this artifact. Let $\mathcal{GB}$, $\mathcal{R}_g$, and $\mathcal{A}$ be respectively the set of current group boards, the set all roles that can be played in a group board created from specification g, and the set of all agents participating to the organisation entity.

- $group\_type(gb, g)$: is true iff the group board $gb \in \mathcal{GB}$ has been created based on the group specification $g \in \mathcal{G}$ (cf. Sec. 3);
- $plays(\alpha, \rho, gb)$: is true iff agent $\alpha \in \mathcal{A}$ plays role ρ in the group board $gb \in \mathcal{GB}$;

The function $rplayers$ returns the number of current players of the role ρ in the group gb.

$$\begin{aligned} &rplayers : \mathcal{R} \times \mathcal{GB} \rightarrow \mathbb{Z} \\ &rplayers(\rho, gb) \stackrel{\text{def}}{=} |\{\alpha \mid plays(\alpha, \rho, gb)\}| \end{aligned} \tag{1}$$

Given the above definitions and the functions $maxrp$ and $minrp$ (cf. Sec. 3), we are able to define the *wellformedness property* of a group

$$\begin{aligned} well\_formed(gb) \leftarrow\ & group\_type(gb, g) \wedge \\ & \forall_{\rho \in \mathcal{R}_g}\ rplayers(\rho, gb) \geq minrp(\rho, g) \wedge \\ & \qquad rplayers(\rho, gb) \leq maxrp(\rho, g) \end{aligned} \tag{2}$$

Since role adoption is the very action that may bring a group in an inconsistent state, two norms bearing on this organisational action are regimented by a GroupBoard: *role compatibility* norm and *role cardinality* norm. In the following norms are represented as a pair. The first argument is the condition part stating when the norm is active. The second argument is the action part stating the obligation, permission, or interdiction.

Role compatibility norm: In the $\mathcal{M}$OISE$^+$ language, roles are incompatible unless explicitly stated the contrary in the organisation specification. When two roles ρ_1 and ρ_2 are specified as compatible inside a group g ($compat(g, \rho_1, \{\rho_2\})$), it implies that an agent that plays ρ_1 in a group board gb created from the specification g cannot perform the operation adoptRole(ρ_i) for any $i \neq 2$ on the corresponding artifact. This constraint on role adoption is formalised by the following norm:

$$\begin{aligned} &(plays(\alpha, \rho, gb)\ \wedge\ group\_type(gb, g)\ \wedge\ compat(g, \rho, C), \\ &\forall_{\rho_i \in \mathcal{R} \setminus C}\ FORBIDDEN_\alpha\ \mathsf{adoptRole}(\rho_i)) \end{aligned} \tag{3}$$

The condition of the norm (the first line) is a conjunction of predicates. Its evaluation is given by the particular status of the group board (that defines whether $plays(\alpha, \rho, gb)$ and $group\_type(gb, g)$ hold or not) and by the structural specification used by the artifact (that defines whether $compat(g, \rho, C)$ holds or not). The action part of the norm (the second line) states that it is forbidden for agent α to execute the action adoptRole on any role that does not belong to the set of compatible roles C. Based on this norm, as soon as an agent adopts a role (activating the norm), the adoption of other roles that are not explicitly stated compatible are forbidden for it.

Role cardinality norm: The number of players of a role in a group is limited by the function $maxrp(\rho, g)$ defined from the Structural Specification. The following norm constrains the role adoption based on the cardinality of the role:

$$(group\_type(gb, g) \ \wedge \ rplayers(\rho, gb) \geq maxrp(\rho, g), \\ \forall_{\alpha \in \mathcal{A}} \forall_{\rho \in \mathcal{R}} \ FORBIDDEN_{\alpha} \ \mathsf{adoptRole}(\rho)) \tag{4}$$

Since these two norms are of the type 'action interdiction', they can be easily implemented in the artifact: whenever the adoptRole operation is triggered by the agent α, the condition of all norms are checked using the structural specification and the current state of the group artifact. If the condition of some of these norms holds, the execution of the corresponding operation is denied.

4.3 SchemeBoard artifact

A SchemeBoard is an organisational artifact providing functionalities to manage the execution of a social scheme. Each SchemeBoard is instantiated upon a specific social scheme specification of the Functional Specification. It coordinates the commitments to missions and the achievement of goals by managing the dependencies between the missions and the goals as described in the social scheme specification. The lifecycle of a SchemeBoard is organised along three phases: formation, goal achievement and finishing. In the formation phase, agents commit to the missions of the scheme. A property of wellformedness conditions the transition to the second phase. A scheme is well-formed if the mission cardinalities are satisfied, i.e. there are enough agents committed to the missions (see below for a more formal definition). In the second phase, goals should be fulfilled by the agents. Each agent is expected to achieve the goals of the missions it is committed to. When the root goal of the scheme is satisfied, the third phase starts and the scheme can be finished and removed from the organisation entity (i.e. the corresponding artifact is destroyed).

During the execution of a scheme, its goals can be in three different states: *waiting*, *possible*, *achieved*. The *waiting* state is the initial state of every goal. In such a state, a goal can not be pursued by the agents. Its change of state depends on the achievement of other goals (called pre-conditions for a goal) or of the wellformedness of the scheme (in case the goal has no pre-conditions). The set of pre-conditions for a goal is deduced from the goal decomposition tree of the scheme. When all pre-conditions of a goal are satisfied and the scheme is well-formed, the state of a goal is changed to *possible*. Then the agent(s) committed to a mission containing that goal can start to achieve it. Let's note that the change from the state *waiting* to *possible* is performed by the SchemeBoard, whereas the change from the state *possible* to *achieved* is performed by the agents.

The observable properties of a SchemeBoard are defined to make an agent able to monitor the overall dynamics concerning execution of the corresponding scheme. It is thus possible for an agent to be aware of which missions are assigned to which agents, which goals are achieved and which can be pursued. Among the observable properties, the most important are the following (Fig. 2):

- OrgBoard: is the reference to the OrgBoard that represents the organisation entity in which the scheme is being executed.
- NormativeBoard: is the reference to the NormativeBoard linked to the scheme [8].
- ResponsibleGroupBoards: contains the references to the GroupBoard that are responsible for the scheme.
- Type: is the identification of the scheme specification in the functional specification (an element of $\mathcal{S}$).
- PlayableMissions: contains all missions that can still be committed to in the scheme.
- PlayersOfMission: contains all the agents committed to a mission of the scheme and their corresponding mission.
- GoalsState: contains the current state of the goals of the scheme.

The usage interface provides the following operations:

- commitMission(m): used by an agent to commit to a mission $m \in \mathcal{M}$;
- leaveMission(m): used by an agent to give up a mission m it is committed to;
- setGoalAchieved(φ): used by an agent to set the state of a goal to *achieved*.

As for the GroupBoard, we define the following predicates bearing on the current state of a SchemeBoard. Let $\mathcal{SB}$ and $\mathcal{M}_s$ be respectively the set of current scheme boards and the set all missions that can be played in a scheme board created from specification s.

- $scheme\_type(sb, s)$: is true iff the scheme board $sb \in \mathcal{SB}$ is created based on the scheme specification $s \in \mathcal{S}$ (the type of a scheme board is defined in its creation);
- $resp\_group(gb, sb)$: is true iff the group $gb \in \mathcal{GB}$ is responsible for the execution of the scheme $sb \in \mathcal{SB}$;
- $committed(\alpha, m, sb)$ is true iff the agent $\alpha \in \mathcal{A}$ is committed to the mission $m \in \mathcal{M}$ in the scheme $sb \in \mathcal{SB}$;
- $achieved(\varphi, sb)$: is true iff the goal φ is already achieved in the scheme sb;
- $possible(\varphi, sb)$: is true iff the state of the goal φ is possible in the scheme sb; considering Φ' the set of all goals that are pre-condition of φ, this predicate can be deduced by

$$possible(\varphi, sb) \leftarrow \bigwedge_{\varphi' \in \Phi'} achieved(\varphi', sb) \ \wedge \ well\_formed(sb) \tag{5}$$

- $succeeded(s)$ it is true that the scheme s has finished successfully.

The function $mplayers$ returns the number of current players of the mission m in the scheme of sb.

$$\begin{aligned} &mplayers : \mathcal{M} \times \mathcal{SB} \rightarrow \mathbb{Z} \\ &mplayers(m, sb) \ \stackrel{\texttt{def}}{=} \ |\{\alpha \mid committed(\alpha, m, sb)\}| \end{aligned} \tag{6}$$

[8] This observable property indirectly links all responsible groups of the scheme to the normative board of the same scheme.

Given the above definitions and the functions $maxmp$ and $minmp$ (cf. Sec. 3), we are able to define the *wellformedness property* of a scheme:

$$well\_formed(sb) \leftarrow scheme\_type(sb, s) \wedge \forall_{m \in \mathcal{M}_s}\ mplayers(m, sb) \geq minmp(m, s) \wedge mplayers(m, sb) \leq maxmp(m, s) \quad (7)$$

Mission commitment norm: Analogously to the *role cardinality* norm, we define a *mission commitment* norm to forbid an agent to commit to missions in a scheme. The number of agents already committed to a mission constrains the action commitMission (mission cardinality). Another constraint is that only agents that play some role in a responsible group for the scheme can commit to a mission in the scheme. We define the mission commitment norm as follows:

$$((scheme\_type(sb, s) \wedge mplayers(m, sb) \geq maxmp(m, s)) \vee (resp\_group(gb, sb) \wedge \neg plays(\alpha, \rho, gb)), FORBIDDEN_{\alpha}\ \mathsf{commitMission}(m)) \quad (8)$$

The implementation of this norm follows the same algorithm used by the GroupBoard: whenever an agent attempts to commit to a mission, if the condition of the norm holds, the operation is denied.

Note that in the current version of ORA4MAS, there is no regimentation on the leaving of a mission or of a role. We consider that these organisational actions should give rise to enforcement and violation. For instance, we could imagine to detect a violation when a mission or a role are left while still having goals of the mission to be achieved. Following the detection of violation, sanctions have to be decided by organisational agents.

5 Organisational Artifacts for Organisation Regulation

Pursuing the description of the basic set of organisational artifacts building ORA4MAS, we turn to the organisational artifacts in relation with the normative dimension of the organisation which is connected to the enforcement mechanisms and to the regulation of the organisational entity. In the current state of the framework, two kinds of such organisational artifacts have been defined: NormativeBoard and ReputationBoard artifacts. They are used to maintain and provide information concerning the agents compliance or not to norms. These artifacts don't provide any operation to the agents since their function is to detect and show as observable properties information related to the current status of the norms given the agents' behaviour related to the groups and scheme they are linked to.

5.1 NormativeBoard artifact

The NormativeBoard artifact (Fig. 2) embeds the functionalities to manage the specification concerning permissions and obligations defined between roles of the

Structural Specification and missions of the Functional Specification. There is one link operation (updateAgentStatus) used by the assigned scheme and groups to trigger an update of the current status concerning a particular agent whenever this agent has performed some operation in the scheme or group.

The norms that are considered in this artifact are not implemented by regimentation, since we would like to allow the agents to violate them. Their implementation is thus not as simple as the implementation of the norms of group and scheme artifacts (where only interdictions are considered and regimentation is used as the mechanism). The NormativeBoard manages the state of the norms as follows (more details are available in [20]). The state of a norm is initially *inactive*. It becomes *active* when its condition holds. When the agent executes the action as it is stated in the action part of the norm, the status of the norm becomes *fulfilled*. In the other case, i.e. the agent does not behave in time accordingly to the action part of the norm, the status of the norm becomes *unfulfilled*.

The set of norms are defined from the deontic specification and the current state of related artifacts. As examples, in the sequel some of these norms are presented.

Obligation to commit to a mission: Based on the deontic relations $obl(\rho, m)$ included in the organisation specification (as defined in section 3), the roles played by the agents (as defined in the section 4.2), and the current number of agents committed to a mission (an agent is not obliged to commit to a mission if the minimum number of players is already achieved), the following norm is defined:

$$\begin{array}{l}(obl(\rho, m) \ \wedge \ plays(\alpha, \rho, gb) \ \wedge \ resp\_group(gb, sb) \ \wedge \\ scheme\_type(sb, s) \ \wedge \ mission\_scheme(m, s) \ \wedge \\ mplayers(m, sb) < minmp(s), \\ OBLIGED_{\alpha} \ \mathsf{commitMission}(m))\end{array} \tag{9}$$

The three first lines of this norm are the condition that states when the norm is active and the last line represents the obligation for the target agent.

Permission to commit to a mission: Based on deontic relations $per(\rho, m)$ included in the organisation specification, and the roles played by the agents, the following a norm which is defined as an interdiction as follows:

$$\begin{array}{l}(\neg(per(\rho, m) \ \wedge \ plays(\alpha, \rho, gb) \ \wedge \ resp\_group(gb, sb) \ \wedge \\ scheme\_type(sb, s) \ \wedge \ mission\_scheme(m, s)), \\ FORBIDDEN_{\alpha} \ \mathsf{commitMission}(m))\end{array} \tag{10}$$

Obligation to achieve a goal: Once an agent α is committed to a mission m, it is obliged to fulfil the possible goals of the mission. The norm below specifies that rule.

$$\begin{array}{l}(committed(\alpha, m, sb) \ \wedge \ goal\_mission(\varphi, m) \ \wedge \ possible(\varphi, sb), \\ OBLIGED_{\alpha} \ \varphi)\end{array} \tag{11}$$

5.2 Artifact for Instrumenting Reputation Processes

Inspired by the concept of *reputation artifact* proposed in [7, p. 101], ORA4MAS is enriched with such a type of artifact in order to provide first class constructs which can be easily used to support the reputation processes. It serves as an indirect sanction instrument for norms enforcement. While direct sanctions are applied when the violation is detected, indirect sanctions have long term results, as is the case of reputation.

This artifact is linked to all the organisational artifacts described in the previous section and to the NormativeBoard artifacts. It can be observed by all agents inside the organisation. The other artifacts notify it about the current state of the organisation. This information is used to compute an *evaluation* for each agent member of the organisation entity. This evaluation is published as an observable property of the artifact. It is important to notice that the evaluation is not the reputation of the agent, as remarked in [7], reputation is a *shared voice* circulating in a group of agents. This artifact is indeed an instrument to influence the reputation of the agent.

Several criteria may be used to evaluate an agent inside an organisation. Herein we chose to evaluate an agent in the context of the roles and missions it is concerned by along three criteria: obedience, pro-activeness, and result.

- *obedience* of an agent is computed by the number of obliged goals it achieves. The goals an agent is obliged to achieve are defined by the deontic specification. All obliged goals that have not been achieved until its deadline are considered as a possible violation (this detection is provided by the normative board). Let's define the following functions: general mission obedience function ($o : \mathcal{A} \rightarrow [0,1]$) and obedience in the context of a particular mission function ($o_m : \mathcal{A} \times \mathcal{M} \rightarrow [0,1]$) and obedience in the context of a particular role ($o_r : \mathcal{A} \times \mathcal{R} \rightarrow [0,1]$). They are computed as follows (in the equations $\#$ is a function that returns the size of a set):

$$o(\alpha) = \frac{\#\{\varphi \mid obliged(\alpha,\varphi) \ \wedge \ achieved(\alpha,\varphi)\}}{\#\{\varphi \mid obliged(\alpha,\varphi)\}}$$

$$o_m(\alpha,m) = \frac{\#\{\varphi \mid obliged(\alpha,\varphi) \ \wedge \ goal\_mission(\varphi,m) \ \wedge \ achieved(\alpha,\varphi)\}}{\#\{\varphi \mid obliged(\alpha,\varphi) \ \wedge \ goal\_mission(\varphi,m)\}}$$

$$o_r(\alpha,\rho) = \frac{\#\{\varphi \mid obliged(\alpha,\varphi) \ \wedge \ goal\_role(\varphi,\rho) \ \wedge \ achieved(\alpha,\varphi)\}}{\#\{\varphi \mid obliged(\alpha,\varphi) \ \wedge \ goal\_role(\varphi,\rho)\}}$$

  $o(\alpha) = 1$ means that the agent α achieved all its obligation and $o(\alpha) = 0$ means it achieved none. $o_m(\alpha,m) = 1$ means that the agent achieved all goals when committed to the mission m, and $o_r(\alpha,\rho) = 1$ means that the agent achieved all goals when playing the role ρ.
- The *pro-activeness* of an agent is computed by the number of goals an agent achieves such that it is not obliged to fulfil that goal in a scheme. The general pro-activeness function ($p : \mathcal{A} \rightarrow [0,1]$) and the pro-activeness in the context

of a particular mission ($p_m : \mathcal{A} \times \mathcal{M} \rightarrow [0,1]$) and role ($p_r : \mathcal{A} \times \mathcal{R} \rightarrow [0,1]$) are defined as follows:

$$p(\alpha) = \frac{\#\{\varphi \mid achieved(\alpha,\varphi) \ \wedge \ \neg obliged(\alpha,\varphi)\}}{\#\Phi \, \#\mathcal{S}}$$

$$p_m(\alpha,m) = \frac{\#\{\varphi \mid achieved(\alpha,\varphi) \ \wedge \ \neg obliged(\alpha,\varphi) \ \wedge \ goal\_mission(\varphi,m)\}}{\#\{\varphi \mid committed(\alpha,m,\_) \ \wedge \ goal\_mission(\varphi,m)\}}$$

$$p_r(\alpha,\rho) = \frac{\#\{\varphi \mid achieved(\alpha,\varphi) \ \wedge \ \neg obliged(\alpha,\varphi) \ \wedge \ goal\_role(\varphi,r)\}}{\#\{\varphi \mid committed(\alpha,m,\_) \ \wedge \ goal\_mission(\varphi,m) \ \wedge \ goal\_role(\varphi,r)\}}$$

$p(\alpha) = 1$ means that the agent achieved all goals it is not obliged to (a highly pro-active behaviour) and $p(\alpha) = 0$ means the contrary.

– The *results* of an agent is computed by the number of successful execution of scheme where it participates. It does not depend on the achievement of the goals in the scheme. It means the agent somehow share the success of the scheme execution and likely has helped for the success. The general results function ($r : \mathcal{A} \rightarrow [0,1]$) and the results in the context of a particular mission ($r_m : \mathcal{A} \times \mathcal{M} \rightarrow [0,1]$) and role ($r_r : \mathcal{A} \times \mathcal{R} \rightarrow [0,1]$) are defined as follows:

$$r(\alpha) = \frac{\#\{s \mid committed(\alpha,\_,s) \ \wedge \ succeeded(s)\}}{\#\{s \mid committed(\alpha,\_,s)\}}$$

$$r_m(\alpha,m) = \frac{\#\{s \mid committed(\alpha,m,s) \ \wedge \ succeeded(s)\}}{\#\{s \mid committed(\alpha,m,s)\}}$$

$$r_r(\alpha,\rho) = \frac{\#\{s \mid committed(\alpha,m,s) \ \wedge \ succeeded(s) \ \wedge \ obl(\rho,m)\}}{\#\{s \mid committed(\alpha,m,s) \ \wedge \ obl(\rho,m)\}}$$

$r(\alpha) = 1$ means that all schemes the agent participated have finished successfully and $r(\alpha) = 0$ means the contrary.

Unlike the previous two criteria, the results value of an agent cannot be increased by the agent itself. This evaluation depends on the performance of all agents committed to the same scheme, creating thus a dependence among them. The selection of good partners is therefore important and the reputation artifact could be used for that purpose.

The aforementioned criteria are combined into a single overall evaluation of an agent ($e : \mathcal{A} \rightarrow [0,1]$) by the following weighted mean:

$$e(\alpha) = \frac{\gamma \, o(\alpha) + \delta \, p(\alpha) + \epsilon \, r(\alpha)}{\gamma \ + \ \delta \ + \ \epsilon}$$

$$e_m(\alpha,m) = \frac{\gamma \, o(\alpha,m) + \delta \, p(\alpha,m) + \epsilon \, r(\alpha,m)}{\gamma \ + \ \delta \ + \ \epsilon}$$

$$e_r(\alpha,\rho) = \frac{\gamma \, o(\alpha,\rho) + \delta \, p(\alpha,\rho) + \epsilon \, r(\alpha,\rho)}{\gamma \ + \ \delta \ + \ \epsilon}$$

The factors γ, δ, and ϵ are used to define the importance of the obedience, pro-activeness, and results values respectively.

All these objective values provided by the reputation artifact can then be used by agents to compute the reputation of others. It is possible that in one organisation where violation is the rule, if you are a strong violator of norms, your reputation is perhaps greater that in an organisation where violation is not at all the rule.

6 Related works and discussion

Several proposals for organisational infrastructures have been proposed in the literature: MadKit, based on AGR organisational model [14]; Ameli [13] and Ameli$^+$ [16], based on Islander [12]; karma, based on TeamCore [30]; OperA [11]; $\mathcal{S}$-$\mathcal{M}$oise$^+$ [23], based on $\mathcal{M}$oise$^+$ [24]. In the sequel, these works and our proposal are discussed considering important topics and features.

Abstraction & encapsulation. The current OIs components are either in agents (Ameli, $\mathcal{S}$-$\mathcal{M}$oise$^+$, OperA) or services (MadKit). The approaches that use only services are not flexible enough to allow the management and change by the agents. Those that use only agents are using them for reactive and task oriented services. Some of those agents are not really pro-active and autonomous entities. In our framework, we raise the level of abstraction with respect to approaches in which organisation mechanisms are hidden at the implementation level. By using agents and artifacts, such mechanisms become parts of the agent world, suitably encapsulated in proper entities that agents then can inspect, reason and manipulate, by adopting a uniform approach.

Agent autonomy. All above mentioned OIs extinguish the agents' autonomy. In Ameli, for instance, the agents are autonomous to achieve goals but the communication is constrained (or regimented) by the OI; in $\mathcal{S}$-$\mathcal{M}$oise$^+$ the agents are autonomous concerning the communication protocols but constrained (or regimented) in the achievement and coordination of collective goals. In our proposal, agents are still autonomous with respect to decision of using or not a specific artifact – including the organisational artifacts – and keep their autonomy – in terms of control of their actions – while using organisational artifacts. Agents however can depend on the functionalities provided (encapsulated) by artifacts, which can concern, for instance, some kind of mediation with respect to the other agents co-using the same organisational artifact. Then, by enforcing some kind of mediation policy an artifact can be both an enabler and a constrainer of agent interactions. However, such a constraining function can take place without compromising the autonomy of the agents regarding their decisions. We also clearly consider two kinds of mechanisms to implement the norms: regimentations that are implemented in the artifacts and can not be violated and enforcement that are implemented both in the artifacts (the detection) and in the organisational agents (evaluation and judgement).

Distributed management. Some OI, as $\mathcal{S}$-$\mathcal{M}$oise$^+$ and MadKit, centralise all the management of the organisation in one agent or service bringing out scal-

ability problems. Distributing the management of the organisation into different organisational artifacts realises a distributed coordination (meaning here more particularly synchronisation) of the different functions related to the management of the organisation. Completing this distribution of the coordination, the reasoning and decision processes which are encapsulated in the organisational agents may be also distributed among the different agents. Thanks to their respective autonomy, all the reasoning related to the management of the organisation (monitoring, reorganisation, control) may be decentralised into different loci of decision with a loosely coupled set of agents.

Openness. To be open to the entrance of heterogeneous agents is an important feature for MAS in general and a reason to establish an organisation for the system. This is thus also an issue considered by all above OIs. In most cases (e.g. $\mathcal{S}$-$\mathcal{M}$OISE$^+$, AMELI) , the agents have access to the organisational infrastructure by means of an agent communication language (KQML, FIPA-ACL) or other open protocols. ORA4MAS does not use a protocol or communication language; operations are used instead. The interaction between the agents and the organisation is no more expressed with an ACL semantic. Besides that, organisational artifacts, as any other kind of artifact, can be created and added dynamically as needed. They have a proper semantic description of both the functionalities and operating instructions, so conceptually agents can discover at runtime how to use them in the best way.
Still related to openness, the approach promotes heterogeneity of agent societies: artifacts can be used by heterogeneous kinds of agents, with different kinds of reasoning capabilities. Extending the idea to multiple organisations, we can have the same agents playing different roles in different organisations, and then interacting with organisational artifacts belonging to different organisations. The use of artifacts, and particularly the CARTAGO implementation, allows agents implemented in different languages to use the artifacts and cooperate using them. Most of the OI listed above give tools and support only for agents implemented in a particular language, normally Java — which is not the most appropriate language to code some types of agents.

'Organisational power back to agents'. The current implementations of OI conceive the organisation as a layer where the application agents relies on to participate in the organisation activities. The agents are not *actors* of this layer, they are simply passive users. This conception of OI is captured by the notion of regimentation and organisation artifacts in our proposal. However, our contribution in this context is to allow that some decisions that were embedded in the services go back to the agents' layer by means of organisational agents. In ORA4MAS artifacts encapsulate the coordination and synchronisation which were implemented in services. Control and judgement procedures are separated from these aspects and are embedded in organisational agents. Organisational agents can then use organisational artifacts to help them in deciding and eventually applying sanctions to other agents.

'Some answers to challenges raised in [1]' In [1] different challenges for building normative multiagent systems have been reported. We attempt in the fol-

lowing to position our work with respect to some of these challenges.
Challenge 1 and *Challenge 2* address respectively the need of tools for agents to support "communities in their task of recognizing, creating, and communicating norms to agents" and tools for agents "to simplify normative systems, recognize when norms have become redundant, and to remove norms". In the framework, the proposal of an OML used to declaratively represent an organisation at the three different levels are a step in the satisfaction of this need. Moreover, by providing OML embedding the expression of norms, these latter are anchored and contextualized within the organisation. Contrary to other approaches which hide the organisation entity, the artifacts building the organisation infrastructure of the framework propose a set of tools to act and manipulate this revealed organisational layer on which the organisation entity is deployed.
Few proposals of enforcement of norms are detailed in the context of an organisational infrastructure. This is also mentioned by the *Challenge 3* "Tools for agents to enforce norms". In the proposed framework, the structuration of the organisational artifacts and agents makes a clear distinction between *enforcement* and *regimentation*. Besides organisational agents, two special kinds of artifacts have been defined to address that challenge: NormativeBoard, ReputationBoard. In the same trend, the distinction complemented by the fact that the framework doesn't modify the agents internal decision proposes a clear basis to address the *Challenge 4* by developping "Tools for agents to preserve their autonomy". Work realised in [5] proposes a good starting point in that direction.

7 Conclusion

In this paper, we have proposed a framework for normative multiagent organisations. It is composed of an organisation modeling language in which norms can be expressed, organisation-awareness mechanisms that are under development and an organisation infrastructure which is based on the A&A meta-model. This latter is composed of a set of organisational artifacts that encapsulate the functional aspects of an organisation and organisation management and regulation. Organisational agents complement this overall picture by encapsulating the decision and reasoning side of the management of organisations and enforcement of norms.

Although we already have some initial results on the use of this framework, some extensions aim at taking advantage of the uniform concepts used to implement the environment and the organisation abstractions through the concept of artifacts. Such an homogeneous conceptual point of view will certainly help us to situate organisations in environment or to install the access to the environment into organisational models (in the same direction as proposed by [27]). Other points of investigation are (1) the study of the reorganisation process of a MAS using the ORA4MAS approach, (2) the impact of the reorganisation on the organisational artifacts, (3) the definition of a meta-organisation for the ORA4MAS, so that we have special roles for organisational agents that give them access to the organisational artifacts.

Acknowledgements

We would like to thank Alessandro Ricci, Michele Piunti and Rosine Kitio for their valuable work within ORA4MAS.

References

1. Guido Boella, Leendert Torre, and Harko Verhagen. Introduction to the special issue on normative multiagent systems. *Autonomous Agents and Multi-Agent Systems*, 17(1):1–10, 2008.
2. Olivier Boissier, Jomi Fred Hübner, and Jaime Simão Sichman. Organization oriented programming from closed to open organizations. In Gregory O'Hare, Michael O'Grady, Oguz Dikenelli, and Alessandro Ricci, editors, *Engineering Societies in the Agents World VII (ESAW 06)*, volume 4457 of *LNCS*, pages 86–105. Springer-Verlag, 2007.
3. Rafael H. Bordini, Jomi Fred Hübner, and Michael Wooldrige. *Programming Multi-Agent Systems in AgentSpeak using Jason*. Wiley Series in Agent Technology. John Wiley & Sons, 2007.
4. Jan Broersen, Mehdi Dastani, Joris Hulstijn, Zisheng Huang, and Leendert van der Torre. The BOID architecture: conflicts between beliefs, obligations, intentions and desires. In Jörg P. Müller, Elisabeth Andre, Sandip Sen, and Claude Frasson, editors, *Proceedings of the Fifth International Conference on Autonomous Agents*, pages 9–16, Montreal, Canada, 2001. ACM Press.
5. Cosmin Carabelea. *Reasoning about autonomy in open multi-agent systems - an approach based on the social power theory*. in french, ENS Mines Saint-Etienne, December 2007.
6. Cristiano Castelfranchi, Frank Dignum, Catholijn M. Jonker, and Jan Treur. Deliberate normative agents: Principles and architecture. In Nicholas R. Jennings and Yves 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 *LNCS*, pages 364–378. Springer, 2000.
7. Rosaria Conte and Mario Paolucci. *Reputation in Artificial Societies: Social Beliefs for Social Order*. Kluwer, 2002.
8. Luciano Coutinho, Jaime Sichman, and Olivier Boissier. Organizational modeling dimensions in multiagent systems. In V. Dignum, F. Dignum, and E. Matson, editors, *Proceedings of Workshop Agent Organizations: Models and Simulations (AOMS@IJCAI 07)*, 2007.
9. Mehdi Dastani. 2APL: a practical agent programming language. *Autonomous Agent and Multi-Agent Systems*, 16:241–248, 2008.
10. Virginia Dignum and Frank Dignum. Modeling agent societies: Co-ordination frameworks and institutions. In Pavel Brazdil and Alípio Jorge, editors, *Proceedings of the 10th Portuguese Conference on Artificial Intelligence (EPIA'01)*, LNAI 2258, pages 191–204, Berlin, 2001. Springer.
11. Virginia Dignum, Javier Vazquez-Salceda, and Frank Dignum. OMNI: Introducing social structure, norms and ontologies into agent organizations. In Rafael H. Bordini, Mehdi Dastani, Jürgen Dix, and Amal El Fallah-Seghrouchni, editors, *Proceedings of the Programming Multi-Agent Systems (ProMAS 2004)*, LNAI 3346, Berlin, 2004. Springer.

12. Marc Esteva, David de la Cruz, and Carles Sierra. ISLANDER: an electronic institutions editor. In Cristiano Castelfranchi and W. Lewis Johnson, editors, *Proceedings of the First International Joint Conference on Autonomous Agents and MultiAgent Systems (AAMAS 2002)*, LNAI 1191, pages 1045–1052. Springer, 2002.
13. Marc Esteva, Juan A. Rodríguez-Aguilar, Bruno Rosell, and Josep L. Arcos. AMELI: An agent-based middleware for electronic institutions. In Nicholas R. Jennings, Carles Sierra, Liz Sonenberg, and Milind Tambe, editors, *Proceedings of the Third International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS'2004)*, pages 236–243, New York, 2004. ACM.
14. Jacques Ferber and Olivier Gutknecht. A meta-model for the analysis and design of organizations in multi-agents systems. In Yves Demazeau, editor, *Proceedings of the 3rd International Conference on Multi-Agent Systems (ICMAS'98)*, pages 128–135. IEEE Press, 1998.
15. Nicoletta Fornara and Marco Colombetti. Specifying and enforcing norms in artificial institutions. In A. Omicini, B. Dunin-Keplicz, and J. Padget, editors, *Proceedings of the 4th European Workshop on Multi-Agent Systems (EUMAS 06)*, 2006.
16. Andrés García-Camino, J.A. Rodríguez-Aguilar, and Wamberto W. Vasconcelos. A distributed architecture for norm management in multi-agent systems. In Jaime Sichman, P. Noriega, J. Padget, and Sascha Ossowski, editors, *Coordination, Organizations, Institutions, and Norms in Agent Systems III*, volume 4870 of *LNAI*, pages 275–286. Springer, 2007. Revised Selected Papers.
17. Benjamin Gâteau, Olivier Boissier, Djamel Khadraoui, and Eric Dubois. Controlling an interactive game with a multi-agent based normative organisational model. In Noriega P., Vázquez-Salceda J., G. Boella, Boissier O., Dignum V., Fornara N., and Matson E., editors, *AAMAS 2006 and ECAI 2006 International Workshops, COIN 2006 Hakodate, Japan, May 9, 2006 Riva del Garda, Italy, August 28, 2006*, volume 4386 of *LNAI*, pages 86–100. Springer, 2007. Revised Selected Papers.
18. Lou Goble and John-Jules Ch. Meyer, editors. *Proceedings of the 8th International Workshop on Deontic Logic in Computer Science, DEON 2006, Utrecht, The Netherlands, July 12-14, 2006*, volume 4048 of *Lecture Notes in Computer Science*. Springer, 2006.
19. Davide Grossi, Huib Aldewered, and Frank Dignum. *Ubi Lex, Ibi Poena*: Designing norm enforcement in e-institutions. In P. Noriega, J. Vázquez-Salceda, G. Boella, O. Boissier, V. Dignum, N. Fornara, and E. Matson, editors, *Coordination, Organizations, Institutions, and Norms in Agent Systems II*, volume 4386 of *LNAI*, pages 101–114. Springer, 2007. Revised Selected Papers.
20. Jomi F. Hübner, Olivier Boissier, Rosine Kitio, and Alessandro Ricci. Instrumenting multi-agent organisations with organisational artifacts and agents: "giving the organisational power back to the agents". *Journal of Autonomous Agents and Multi-Agent*, 2009. To Appear.
21. Jomi Fred Hübner. *Um Modelo de Reorganização de Sistemas Multiagentes*. PhD thesis, Universidade de São Paulo, Escola Politécnica, 2003.
22. Jomi Fred Hübner, Olivier Boissier, and Laurent Vercouter. Instrumenting multi-agent organisations with reputation artifacts. In Jomi F. Hübner, Eric Matson, Olivier Boissier, and Virginia Dignum, editors, *Coordination, Organizations, Institutions, and Norms in Agent Systems IV*, volume 5428 of *LNAI*, pages 96–110. Springer, 2009.
23. Jomi Fred Hübner, Jaime Simão Sichman, and Olivier Boissier. S-MOISE+: A middleware for developing organised multi-agent systems. In Olivier Boissier, Virginia

Dignum, Eric Matson, and Jaime Simão Sichman, editors, *Coordination, Organizations, Institutions, and Norms in Multi-Agent Systems*, volume 3913 of *LNCS*, pages 64–78. Springer, 2006.

24. Jomi Fred Hübner, Jaime Simão Sichman, and Olivier Boissier. Developing organised multi-agent systems using the MOISE+ model: Programming issues at the system and agent levels. *International Journal of Agent-Oriented Software Engineering*, 1(3/4):370–395, 2007.
25. Rosine Kitio, Olivier Boissier, Jomi Fred Hübner, and Alessandro Ricci. Organisational artifacts and agents for open multi-agent organisations: "giving the power back to the agents". In Jaime Sichman, P. Noriega, J. Padget, and Sascha Ossowski, editors, *Coordination, Organizations, Institutions, and Norms in Agent Systems III*, volume 4870 of *LNCS*, pages 171–186. Springer, 2008. Revised Selected Papers.
26. M. Luck, P. McBurney, O. Shehory, and S. Willmott. *Agent Technology: Computing as Interaction (A Roadmap for Agent Based Computing)*. AgentLink, 2005. `http://www.agentlink.org/roadmap`.
27. 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, *Coordination, Organizations, Institutions, and Norms in Agent Systems II*, volume 4386 of *LNAI*, pages 133–146, 2007.
28. M. Piunti, A. Ricci, L. Braubach, and A. Pokahr. Goal-directed interactions in artifact-based mas: Jadex agents playing in cartago environments. In *IEEE/WIC/ACM Conferences on Web Intelligence and Intelligent Agent Technology (IAT-2008)*. IEEE/WIC/ACM, 2008.
29. Alexander Pokahr, Lars Braubach, and Winfried Lamersdorf. Jadex: A BDI reasoning engine. In Rafael H. Bordini, Mehdi Dastani, Jürgen Dix, and Amal El Fallah Seghrouchni, editors, *Multi-Agent Programming: Languages, Platforms, and Applications*, number 15 in Multiagent Systems, Artificial Societies, and Simulated Organizations, chapter 6, pages 149–174. Springer, 2005.
30. David V. Pynadath and Milind Tambe. An automated teamwork infrastructure for heterogeneous software agents and humans. *Autonomous Agents and Multi-Agent Systems*, 7(1-2):71–100, 2003.
31. Alessandro Ricci, Michele Piunti, L. Daghan Acay, Rafael H. Bordini, Jomi F. Hübner, and Mehdi Dastani. Integrating heterogeneous agent programming platforms within artifact-based environments. In Lin Padgham, David C. Parkes, Jörg Müller, and Simon Parsons, editors, *7th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2008), Estoril, Portugal, May 12-16, 2008*, pages 225–232. IFAAMAS, 2008.
32. Alessandro Ricci, Mirko Viroli, and Andrea Omicini. CArtAgO: A framework for prototyping artifact-based environments in MAS. In Danny Weyns, H. Van Dyke Parunak, and Fabien Michel, editors, *Environments for MultiAgent Systems III*, volume 4389 of *LNAI*, pages 67–86. Springer, May 2007. 3rd International Workshop (E4MAS 2006), Hakodate, Japan, 8 May 2006. Selected Revised and Invited Papers.
33. Alessandro Ricci, Mirko Viroli, and Andrea Omicini. The A&A programming model & technology for developing agent environments in MAS. In Mehdi Dastani, Amal El Fallah-Seghrouchni, Alessandro Ricci, and Michael Winikoff, editors, *Programming Multi-Agent Systems, 5th International Workshop, ProMAS 2007,*

Honolulu, HI, USA, May 15, 2007, Revised and Invited Papers, volume 4908 of *LNCS*, pages 89–106. Springer, 2008.
34. Raimo Tuomela and Maj Bonnevier-Tuomela. Norms and agreement. *European Journal of Law, Philosophy and Computer Science 5*, pages 41–46, 1995.
35. J. Vázquez-Salceda, H. Aldewereld, and F. Dignum. Norms in multiagent systems: some implementation guidelines. In *Proceedings of the Second European Workshop on Multi-Agent Systems (EUMAS 2004)*, 2004. `http://people.cs.uu.nl/dignum/papers/eumas04.PDF`.