Artifacting and Regulating the Environment of a Virtual Organization

Sergio Esparcia*, Roberto Centeno†, Ramón Hermoso‡ and Estefanía Argente*
*Grupo de Tecnología Informática - Inteligencia Artificial
Universitat Politècnica de València, Spain
{sesparcia, eargente}@dsic.upv.es
†Departamento de Lenguajes y Sistemas Informáticos
Universidad Nacional de Educación a Distancia, Madrid, Spain
rcenteno@lsi.uned.es
‡Centre for Intelligent Information Technologies (CETINIA)
University Rey Juan Carlos, Madrid, Spain
ramon.hermoso@urjc.es

***Abstract*—This work presents an extension of the Environment Dimension of the Virtual Organization Model, which is an Organization Modeling Language to define Organization-Centered Multi-Agent Systems. This extension allows this model to regulate the environment by supporting *artifacts for organizational mechanisms*, an approach based on the Agents & Artifacts conceptual framework. The three main entities of this framework are agents, artifacts and workspaces, which have been integrated in this work inside the Virtual Organization Model. Additionally, this paper presents a case study in the health care domain using this approach and an analysis of the related work on this topic.**

***Keywords*-multi-agent systems, virtual organizations, artifacts, organizational mechanisms, environment, agent-oriented software engineering**

I. Introduction

Open Multi-Agent Systems (OMAS) are characterized by being designed with a general purpose in mind, and having an unknown population of autonomous agents, at design time. Due to their open nature, the general problem when designing OMAS consists on assuring that agents will behave according to the system's objectives and preferences. This problem has been tackled by endowing such systems with organizational abstractions such as roles, norms, tasks, etc. in charge of regulating the behavior of the population.

In these systems, the environment [1] appears as a key concept since it restricts the interactions between external agents and the own system. Therefore, changes in the environment can be introduced so as to influence agents' behavior towards "better" situations from the system's point of view [2]. Besides, when we focus on OMAS regulated by organizational concepts (Organization-Centered MAS, OCMAS) [3], the own environment helps limiting the internal relationships between agents and resources available to be used by the population.

When defining software systems, designers can use metamodels [4], which are a mechanism that allows defining modeling languages in a formal way, establishing the primitives and the syntactic-semantic properties of a model. Organizational Modeling Languages (OML) [5] are metamodels used by OCMAS designers to define the elements that the system will contain at runtime. These OMLs extend existing modeling languages to include organizational elements. OMLs enable modeling agent coordination inside open systems and establishing mechanisms that control the organization in a social level, i.e. interactions between agents, organizational goals, norms, etc. [6]. These models include both individual and global perspectives, and few of these proposals try to provide models capable of representing organizational change, in order to give response to changes in the environment or in the organizational structure [7].

Some approaches have been presented with the aim of modeling and improving the environment of a Multi-Agent System (MAS), being the Agents & Artifacts (A&A) conceptual framework [8] one of the most recognized. This proposal models the environment around three main concepts: (i) agents, the proactive entities of the system; (ii) artifacts, passive entities that help agents to reach their objectives by means of their provided functionality; and (iii) workspaces, used to model the topology of the environment.

Other proposals rely on the statement that the environment can be used to modify the behavior of a MAS from both a macro perspective (from the system's point of view) and a micro perspective (from agents' point of view). For example, Centeno *et al.* [9] put forward that organizational mechanisms can be introduced in the environment with the aim of influencing agents' behavior towards more effectiveness with regard to the global purpose of the system.

Nevertheless, most of the current approaches suffer from not taking into account the design of environmental regulation at design time. They usually focus on the design of entities that form part of the system but they do not pay much attention on how those entities should be regulated in order to better achieve the system's goals. Thus, allowing the design of mechanisms for regulating the environment becomes a non-straightforward task. Many models have specific abstractions to allow certain kind of regulations, such as norms, policies, etc., but they lack of more generic and reusable design-time entities that facilitate the regulation of the environment.

The objective of this work is to extend a previously validated OML to design OCMAS, by adding first-class abstract entities that allow modeling the regulation of the environment, since there are no OMLs that provide these mechanisms. In particular: (i) artifacts for organizational mechanisms theory provides concepts that help designers to model the environment [10]; (ii) we intend to enable regulation of the environment by using artifacts for organizational mechanisms abstractions; and (iii) also provide reusable first-class abstractions to build regulating mechanisms at design time.

We have chosen the existing Virtual Organization Model (VOM) [11] as a modeling language to be extended since it was developed inside the *Grupo de Tecnología Informática - Inteligencia Artificial* research group from *Universitat Politècnica de València*, where two of the authors of this work are currently developing their work. VOM is also providing support to the Agent-Oriented Software Engineering methodology GORMAS [12], which provides a set of guidelines and patterns to develop virtual organizations. Therefore, we have such high level of knowledge about VOM, so introducing modifications inside it would be easier than modifying a different OML. VOM will be modified so as to support concepts such as artifacts and organizational mechanisms that help describing and operating with the environment. As starting point, the work presented in [10] is taken into account, where both proposals, artifacts and organizational mechanisms, were merged to define Artifacts for Organizational Mechanisms, which are provided with functionalities that modify and improve the global behavior of a MAS.

The rest of this work is structured as follows: Section II describes both Artifacts for Organizational Mechanisms, and the Virtual Organization Model. Section III presents the new environment dimension for the Virtual Organization Model that integrates Artifacts for Organizational Mechanisms. Section IV presents how a real application, in the domain of health care, could be modeled by using our proposal. Section V depicts some related work and compares it with our proposal. Finally, Section VI puts forward some conclusions and future work.

II. Background

This section describes the two basic concepts that this proposal is based on. On the one hand, we put forward a work that merges the concepts of artifacts and organizational mechanisms, the *artifacts for organizational mechanisms* [10]. On the other hand, the remainder of the section will describe the Virtual Organization Model [11], which is the Organizational Modeling Language (OML) chosen for this work, as previously explained.

A. Artifacts for Organizational Mechanisms

In order to cope with OCMAS environments, it is necessary to come up with tools that facilitate this task. For this reason, the *Artifacts for Organizational Mechanisms* [10] were developed. They model the *Organizational Mechanisms* [9] by means of artifacts from the *Agents & Artifacts* (A&A) conceptual framework [8]. Organizational Mechanisms are introduced in a MAS in order to influence agents' behavior towards more effectiveness regarding some goals from both a macro and a micro perspective. Hence, these mechanisms can provide additional information to agents that may persuade (or incite) them to behave in a certain way, so adjusting their behavior to system's goals.

Organizational mechanisms have been modeled as artifacts to facilitate developers to better deploy and implement them, as well as adding functionality in MAS environments. Three types of Artifacts for Organizational Mechanisms have been defined: (i) *Informative Artifacts*, which provide information to an agent, based on the internal state of this agent and the partial view of the environment that the artifact has; (ii) *Incentive Artifacts*, which modify the global behavior of the system by changing the incentive system of the MAS; and (iii) *Coercive Artifacts*, which update the action space of an agent. All these artifacts make use of the environment of a MAS, so they can exploit all knowledge they have about entities populating the system.

B. Virtual Organization Model

This section briefly describes the Virtual Organization Model (VOM) [11]. This OML follows a UML approach, and it is aimed to model Organization-Centered MAS, identifying the common elements that are present in an organization. As most of the metamodels, VOM also gives support to a software development methodology by upholding the development of the Virtual Organizations defined in GORMAS [12] methodology. VOM is composed of five organizational dimensions: structural, functional, dynamical, environment and normative. Systems defined by VOM are structured by means of the elements identified in these dimensions:

- **Structural Dimension**. Describes the components of the system and their relationships. It defines the organization, composed of agents and organizational units, roles, and their social relationships.
- **Functional Dimension**. Details the functionality of the system based on services, tasks and objectives. It also describes the stakeholders that interact with the organizational units, the services offered by the organization, and the resources used by the organization.
- **Dynamical Dimension**. Defines interactions between agents, as well as the role enactment process, defining the roles that organizational units or agents are able to play.

- **Environment Dimension**. The environment of the organization is defined by means of resources and applications that the organization has available. Also, ports are defined in this dimension. They are entities that control the access to resources and applications, and are additionally used to publish services.
- **Normative Dimension**. Describes the normative restrictions over the behavior of the entities that populate the system, including organizational norms that agents must fulfil, with associated sanctions and rewards.

Figure 1 depicts the Environment Dimension of VOM. The main entities represented in this model are Resources and Applications. *Resources* are environmental objects that do not provide a specific functionality, but they are essential for the task execution. *Applications* are employed to model passive services, i.e. a set of operations that do not require agent interaction to be provided and executed. Applications and resources are contained in organizational units (i.e. they can be seen by members of the organization) or they belong to agents. Access to applications and resources is carried out by means of *Environment Ports*, whereas *Service Ports* are employed for publishing services and controlling their access. Thus, a system designer can specify who is responsible for the management of each port, and also who will have reading and writing permissions over the resource, application or service that each port is responsible of.

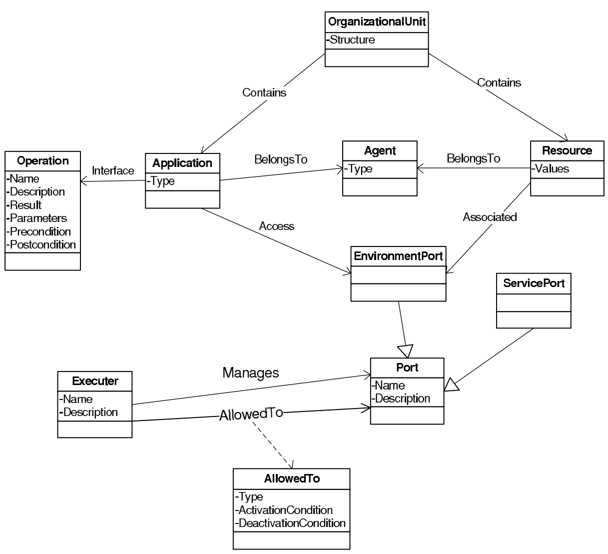

Figure 1. Environment Dimension of the VOM metamodel

This Environment Dimension is mainly focused on describing the elements of the environment (e.g. resources, agents and organizational units), the operations that can be made over those elements (e.g. applications and services) and who is enabled to make use of those elements (e.g. by using ports). However, no mechanisms for regulating the environment can be easily defined, such as incentive or coercive mechanisms that modify the action space of agents. If needed, systems designers should try to define them by means of applications, norms, services or other elements not directly related with the environment, being completely lacking of guidelines from the metamodel.

Therefore, as previously explained, the main objective of this work is to integrate Artifacts for Organizational Mechanisms into the Virtual Organization Model so as to better assist the system designer when modeling the environment. This proposal will be presented in the following section.

III. Artifacting the environment

This section presents an extension of the Environment Dimension of VOM (see Figure 1) in order to improve the way that the environment is regulated in a VOM-defined MAS. This extension has been carried out in two different activities:

1) **Integrating applications, resources, and ports by means of artifacts.** The main objective of adding artifacts in VOM is to have available all the operations and observable properties of the three types of Artifacts for Organizational Mechanisms (informative, incentive and coercive artifacts). Additionally, artifacts present features (which will be enumerated later in this section) that encapsulate and extend functionalities provided by the resources, applications and ports of the Environment Dimension.
2) **Clearly representing where environmental elements are located by using the concept of workspace.** Workspaces will help designers to physically structure the environment, so each entity inside the environment will have a specific location.

In the first activity, resources, applications and ports have been integrated inside the artifact concept as follows: (i) **Resources** are entities that are provided with a set of values that can be checked, reduced and/or increased by agents. Since artifacts provide observable properties that can represent these values, resources have been replaced by artifacts in the new proposal of the Environment Dimension. Note that to modify these properties, a set of operations can be established inside an artifact. (ii) **Applications**, which are interfaces for operations, can be implemented by means of artifacts since their operations can be directly translated into the artifact's operations. (iii) The access management of resources and applications carried out by the **Environment ports** can be handled now by artifacts, which are provided with an *internal state* and a *function description* (that acts as an operating manual) where the information about permissions is stored. For **Service ports**, the internal state and function description of an artifact are also employed to control permissions whereas services are published by means of the observable properties of the artifact.

In the second activity, the Environment Dimension of VOM is provided with a physical description of the environment by means of the **workspace** concept, in which artifacts are located. For each workspace, an absolute position (named location) is given. Workspaces of the system can be intersected and nested, features that allow artifacts to be placed in different workspaces, so they are able to be perceived and used from various locations. As in the original Environment Dimension, Agents and Organizational Units are related to entities from this dimension. In this case, an agent is able to perceive a set of workspaces, and use the artifacts that are located inside this set. Additionally, an organizational unit can be placed inside a workspace, so this organizational unit has a specific location inside the organizational environment.

It should be noted that the organization is represented as the whole set of organizational units that make it up. Moreover, the environment of a system is the aggregation of all the workspaces of the system, where the organization to be designed will be located. The addition of workspaces makes the designer able to model the environment in a similar way than the real-world environment would be designed. The previous version of VOM was not able to explicitly define the visibility or the particular location of each element inside the environment. Using our proposed extension, designers can now clearly define these features.

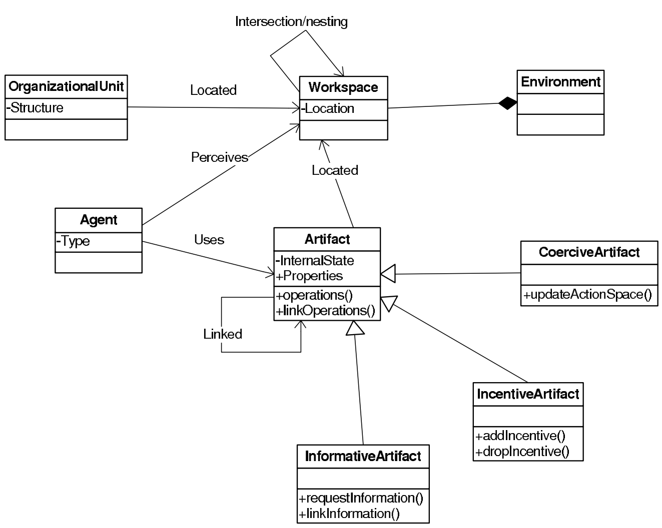

Figure 2. *New Environment Dimension including artifacts and workspaces*

As depicted in Figure 2, the new version of the Environment Dimension is now structured around the *Artifact* entity. There are three different types of artifacts, representing the three types of Artifacts for Organizational Mechanisms, inheriting from the main artifact entity. Each type of artifact is represented in the metamodel by means of its particular operations and observable properties. Informative artifacts are provided with operations that allow agents (and other artifacts) to request information. Incentive artifacts, whose goal is to modify the reward system of the MAS, are enhanced with operations for adding and deleting incentives from this reward system. Finally, coercive artifacts are able to modify the action space of an agent by means of its particular operation.

These modifications enhance the metamodel by making it a proper coordination system for a Virtual Organization, and improving the way how the environment is regulated. As explained before, in the previous version of VOM, the environment could have been regulated by means of applications, services or norms, but since these elements were not defined with this objective in mind, introducing mechanisms for regulating a system might imply a rather hard work for designers. With this new addition, environmental regulations can be explicitly developed using Artifacts for Organizational Mechanisms (whose access permissions are controlled by the artifacts themselves, being not necessary to use ports). Features and functionalities from this type of artifacts were presented in a previous work [10] in a formal approach, but a specific deployment of this proposal in a software development approach was not provided. By including Artifacts for Organizational Mechanisms inside the Environment Dimension of VOM, the means for introducing them inside an agent-based system has been provided here.

IV. Case Study

In this section we focus on a real problem in the health care domain. The domain of medical assistance, in general, includes many tasks that require flexible on-demand negotiation, initiation, coordination, information exchange and supervision among different involved entities (e.g. ambulances, emergency centers, hospitals, patients, physicians, etc.). In particular, we focus on coordination of inter-hospital transfers. This task is performed by the SUMMA112, which is the emergency center in charge of providing sanitary assistance to urgencies, emergencies, catastrophes and special situations in the Autonomous Region of Madrid, in Spain. The aim of this task is to coordinate the transfer of patients among different hospitals, for example, in order to provide a specific treatment.

The solution adopted by the SUMMA112 is currently based on an agreement process between the SUMMA112 and the hospitals that have enough resources to treat the patient and, at the same time, they want to treat the patient. That is, sometimes hospitals have the resources to treat the patient, but for any reason they are not interested in taking this kind of patient, e.g. because they are carrying out a clinical trial regarding patients suffering other kind of disease.

A. *Proposed solution*

Once the problem definition has been presented, we introduce a possible solution to such a problem based on Artifacts for Organizational Mechanisms. Following an agent-based approach, each hospital could be modeled as an autonomous agent participating in the system whose objective is to successfully plan the inter-hospital transfer. Besides, the system's administrator - in our case it could be the own SUMMA112 coordination center - is able to endow the environment with organizational artifacts, aiming to fulfil the global objective.

We tackle the problem by relying on an incentive artifact coupled with an informative one. The objective of the *incentive artifact* is to modify consequences of some actions so as to agents have incentives to perform (or not) such actions. In this case, the coordination center needs hospitals to perform the action `admit patient`. Thus, since the coordination center knows that hospitals are interested in having a high budget, consequences of such an action will be modified in a way that the hospital's budget will be increased when it performs that action. This fosters the positive behavior of hospitals but does not keep them from not accepting patients they are not interested in. Thus, the artifact should also try to maximize the number of positive bids for accepting a patient when a transfer is needed, by imposing sanctions for not accepting patients, e.g. by reducing budget in some medical services.

Moreover, in order to allow agents to reason about rewards added to the system, there exists an *informative artifact*. In this case, the artifact will provide the consequences (rewards) of the action `admit patient`.

Any transfer negotiation process has only one "winner", what means that only one hospital admits the patient. Facing this fact, a question raises: among the remainder hospitals that have not "won" the patient, which of them should be punished? It seems clear that a difference should be made between those who bid for the patient and those who did not. Even if the hospital really wants to accept the patient but it does not get him/her, would not be considered the same as when a hospital is not interested in the patient and does not bid for him/her. Some other non-desirable cases could happen. For instance, a hospital that bids for a patient, but with so low level of interest. Therefore, this hospital previously knows that the patient will not be assigned to it. This last case cannot be avoided by using incentive artifacts. For that reason, we propose to add a *coercive artifact* to the system to keep hospitals from not accepting patients in which they are not interested in. The objective of coercive artifacts lies in the need of producing changes in the environment of the system by producing changes in the agents' capability functions. In this case, we suggest allowing the artifact to close certain medical services in hospitals that do not bid for patients that could be potentially treated by those same services. Closing services is equivalent to ban the action of accepting patients for that service in that hospital.

Additionally, it should be noticed that the private and personal data of the patients is not jeopardized by putting this solution into practice, since artifacts are provided with read and use permissions. Therefore, artifacts are able to control who could access to their delicate information.

B. Example

Let us show an example of this problem. Juan Domínguez, 23 years old, is waiting for assistance in the *Hospital de El Escorial*,[1] where he has been detected to have appendicitis. In this case, laparoscopic surgery is required. Juan would need to stay for another two days under observation, and would have to come again two weeks later for cure. After the diagnosis, the physicians decided that he should be treated in the next 24 hours. However, because of an excess of operations at this time in that hospital, there is no means to provide the appropriate treatment in this hospital. Thus, Juan should be transferred to another hospital that has sufficient capacities to treat his case.

Management at *Hospital de El Escorial* calls the SUMMA112 coordination center to request the transfer. Therefore, the center is in charge of providing a solution to such a problem by assigning a destination hospital to that patient. Different alternatives to transfer the patient are feasible. There are two available hospitals to do that: *Hospital de Móstoles* and *Hospital Reina Sofía*. Therefore, a negotiation table is formed to carry the transfer out. This table is composed of hospitals' representatives (*Agent-1* and *Agent-2* for *Hospital de Móstoles* and *Hospital Reina Sofía* respectively), and *Agent-cc* representing SUMMA112.

VOM is used in order to model the solution, representing all the participating entities. VOM uses GOPPR [13] notation to represent its entities. Figure 3 depicts a caption including all the entities that will be used to model this case of study.

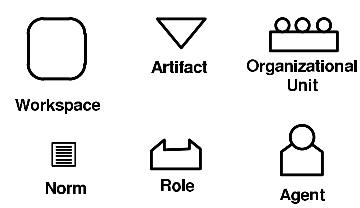

Figure 3. Caption for some of the VOM entities

Prior to the modeling of the environment, "Plays" relationships between roles, and agents and organizational units should be represented inside the Dynamical Dimension of VOM (see Figure 4). In this dimension, features such as interactions between agents or the role enactment process are described. In our scenario, hospitals, the negotiation table, Madrid's Health Care System and SUMMA112 are modeled by means of organizational units.

Using these entities from VOM, we are able to model the environment of this case of study. As it can be seen in Figure 5, the main entity of this diagram is an organizational unit

[1]It is a hospital located in the north of the region of Madrid

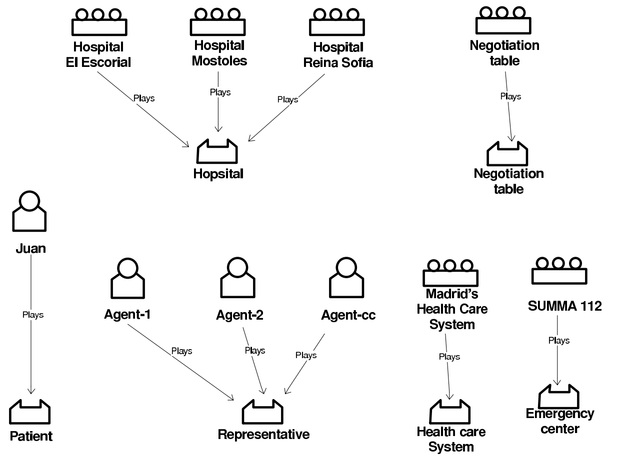

Figure 4. Dynamical Dimension, depicting organizational units, agents and roles

representing the health care system of the region of Madrid. Furthermore, let us suppose that this Organizational Unit (OU) is containing other OUs representing the hospitals that participate in this scenario (*Hospital de El Escorial*, *Hospital de Móstoles* and *Hospital Reina Sofía*), the SUMMA112 service and the negotiation table. Each hospital, and the SUMMA112 service, are provided with a representative agent. Also Juan, the patient, which is the main reason for creating this scenario, is represented by means of an agent entity.

The required artifacts for organizational mechanisms are located inside the same workspace as the OU that represents the health care system (see Figure 5). Thus, agents belonging to the Madrid's Health Care System can make use of these artifacts. More specifically, representative agents are allowed to use informative artifacts to obtain the information they require during this process. Since the coordination center is the administrator of the system, it has enough permissions to operate directly with the incentive and coercive artifacts, in order to introduce incentives and sanctions into the negotiation process. Additionally, incentive and coercive artifacts are linked to the informative artifact since they can send information that will be incorporated into the internal state of the informative artifact, making this information available to representative agents.

It should be noted that this model only represents the interactions at design time, and it does not represent changes at runtime. Thus, in this example, agents *Agent-1* and *Agent-2* (representatives of the hospitals) and *Agent-cc* from SUMMA112 will have a meeting inside the Negotiation Table OU, thus they will have to move from their own OUs to the Negotiation Table.

Table I gives a description of the different functions implemented inside the artifacts that are present in this example.

The proposed solution points out some advantages of using the VOM model extended with Artifacts for Organizational Mechanisms. In order to develop the same domain (and scenario) with the previous Environment Dimension of VOM (i.e. without the extension), regulation would have had to be achieved by using *application* and *resource* entities to model regulation processes and *ports* in order to enable access to them. With the proposed model extension, oriented to regulation design, the artifacts for organizational mechanisms allow an easier design, since they permit to deploy any kind of regulating mechanisms for the system from a common interface for different types of regulation mechanisms (informative, incentive or coercive).

V. Organizational Modeling Languages

In this section, different OMLs that explicitly model the environment of an OCMAS are reviewed. More specifically, this section will focus on the following proposals: GAIA [14], OMACS [15], SODA [16], CArtAgO [17] and MEnSA [18].

GAIA [14] is the first OML that explicitly took into account social concepts. The environment of a MAS designed by GAIA is defined by means of the resources that roles have available. Each resource has a set of associated actions that is able to execute. The portion of the environment that agents can sense and effect is determined by their specific role and current status. The previous version of VOM was provided with the resource concept taken from Gaia, but in this new proposed version resources and their properties are embedded into the concept of artifact.

The **OMACS** [15] metamodel provides suitable mechanisms that allow the system to reorganize itself at runtime. OMACS domain model consists on a set of *environmental objects* that describe the elements of the system and a set of *properties* that specify the principles and processes that govern the environment.

Artifacts and workspaces were firstly introduced in an OML by the **SODA** metamodel [16], whose most recent version is compliant with the A&A (Agents & Artifacts) conceptual framework. The environment is thus modeled and structured by means of artifacts and workspaces.

CArtAgO [17] is a framework built following the principles of the A&A conceptual framework. Its metamodel is composed of three main entities: (i) *agent bodies*, which make it possible to situate agents inside the environment; (ii) *artifacts* as the basic building blocks for structuring the environment; and (iii) *workspaces* aimed at defining the topology of the environment. This metamodel can be considered as a 'light' version of the SODA metamodel, but focusing on implementation details. Similarly to SODA, this metamodel enables describing the components of the environment but not how to regulate them.

MEnSA metamodel [18] integrates fragments from well known metamodels: Gaia, PASSI [19], SODA and Tropos

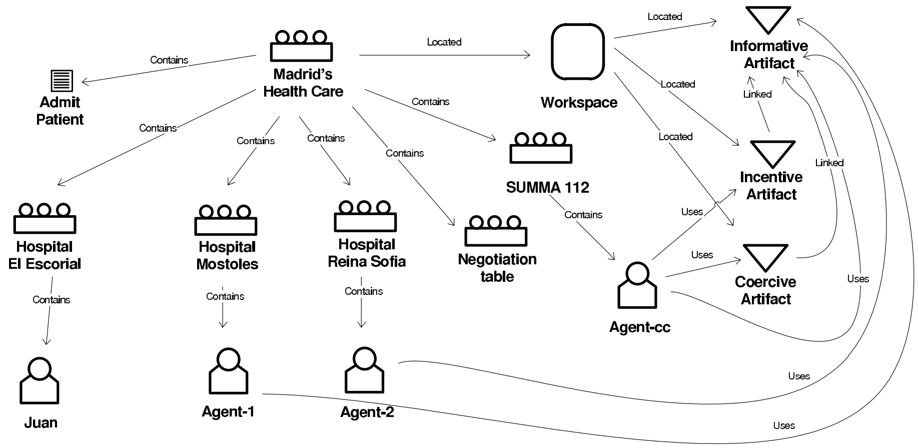

Figure 5. Environment Dimension of our case of study

Functions of the Artifacts for Organizational Mechanisms	
Name	**Description**
Informative Artifact	
New Patient	Inserts the information about a new patient in the internal state of the artifact
Add Reward Information	A link operation that inserts the information about a new reward in the internal state of the artifact
Apply Incentive	Inserts the information about a new incentive being applied over an artifact in the internal state of the informative artifact
Write Argument	Inserts an argument from an agent in the internal state of the artifact
Get Action Consequences	Returns the information about carrying out an action
Get Common Sets	Returns the common sets computed after an argumentation process
Add Constraints	Inserts the information about a new constraint being applied in the internal state of the informative artifact
Incentive artifact	
Add Incentive	Introduces a new incentive in the reward system of the organization
Coercive artifact	
Update Action Space	Updates the action space of an agent by allowing or forbidding him to carry out actions
Apply Constraint	Applies a constraint over a hospital of the organization

Table I
OPERATIONS THAT ARE IMPLEMENTED IN THE ARTIFACTS FOR ORGANIZATIONAL MECHANISMS OF THE CASE STUDY

[20]. It is divided into requirements, design and implementation steps. In the *requirements* and *design* steps the topology is defined through *workspaces* that build the environment, in which *artifacts* are located (like in SODA). Artifacts can also be grouped by means of compositions.

Neither of these OMLs provides mechanisms nor facilities for regulating the environment. We could have selected any of them as a basis for our extension, specially those ones that employ the artifact concept; but we decided to extend VOM since we have a great knowledge of this OML and it gives support to GORMAS [12] methodology, which provides a set of guidelines that helps designers to model organizations and employs a softwared engineering method combined with an engineering process for designing human organizations based on the Organizational Theory [21].

VI. Conclusions and Future Work

In this work, the Environment Dimension model of the Virtual Organization Model [11] has been extended by endowing this model with Artifacts for Organizational Mechanisms introducing the means for regulating the environment. Besides, the concept of *workspace* has been introduced as the means to define the topology of the environment, helping in the distribution of agents and artifacts inside OCMAS, in a similar way that SODA and MEnSA metamodels do.

In particular, informative, incentive and coercive artifacts have been added to the metamodel. Informative artifacts are passive entities used by agents providing them with some information in order to help them in their deliberative process. Incentive artifacts are introduced with the aim of modifying the consequences of actions such that agents have incentives (sanctions and rewards) to perform a particular action. Finally, coercive artifacts are aimed to produce changes in the environment of the system by producing changes in the agents' capability functions, adding or removing actions to the agents' action space.

With the addition of the artifacts, and especially with the addition of the Artifacts for Organizational Mechanisms, the resulting extended metamodel becomes a useful coordination tool for an Organization-Centered Multi-Agent System since they introduce regulations into the environment that will positively affect the global utility of an OCMAS. Both artifacts and organizational mechanisms provide features that can help agents to coordinate and organize a system without the need of other methods and tools. Additionally, with the addition of workspaces, it is possible to physically define the environment, thus giving an absolute location to each entity populating this environment.

As a future work, we are working on the introduction of artifacts into the THOMAS architecture [22], so then being able to develop artifacts not only from a design point of view, but also carrying out the implementation of this important enhancement of a MAS. Moreover, a library of artifacts, defining patterns for Artifacts for Organizational Mechanisms that facilitate their development will be implemented.

Acknowledgment

This work is supported by TIN2009-13839-C03 and PROMETEO/2008/051 projects of the Spanish government, CONSOLIDER-INGENIO 2010 under grant CSD2007-00022, and the COST Action IC0801.

References

[1] D. Weyns and H. Parunak, "Special issue on environments for multi-agent systems," *Autonomous Agents and Multi-Agent Systems*, vol. 14, no. 1, pp. 1–116, 2007.

[2] R. Centeno, H. Billhardt, and S. Ossowski, "Inducing desirable behaviour through an incentives infrastructure," in *Proceedings of the 8th German conference on Multiagent system technologies*, ser. MATES'10. Springer-Verlag, 2010, pp. 64–75. [Online]. Available: http://portal.acm.org/citation.cfm?id=1887504.1887515

[3] C. Lemaître and C. Excelente, "Multi-agent organization approach," in *Proc. II Iberoamerican Workshop on DAI and MAS*, 1998.

[4] J. P. van Gigch, *System Design Modeling and Metamodeling*. Plenum Press, 1991.

[5] E. Argente, G. Beydoun, R. Fuentes-Fernández, B. Henderson-Sellers, and G. Low, "Modelling with Agents," *Agent-Oriented Software Engineering X*, pp. 157–168, 2011.

[6] V. Dignum, J. Meyer, H. Wiegand, and F. Dignum, "An organization-oriented model for agent societies," in *Proc. Int. Workshop on Regulated Agent-Based Social Systems (RASTA-02)*, 2002.

[7] V. Dignum and F. Dignum, "A landscape of agent systems for the real world," Institute of Information and Computing Sciences, Utrecht University, Tech. Rep. 44-CS-2006-061, 2006.

[8] A. Ricci, M. Viroli, and A. Omicini, "Give agents their artifacts: the a&a approach for engineering working environments in mas," in *Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*. ACM, 2007, pp. 150–152.

[9] R. Centeno, H. Billhardt, R. Hermoso, and S. Ossowski, "Organising MAS: a formal model based on organisational mechanisms," in *Proceedings of the 2009 ACM symposium on Applied Computing*. ACM, 2009, pp. 740–746.

[10] S. Esparcia, E. Argente, R. Centeno, and R. Hermoso, "Enhancing MAS Environments with Organizational Mechanisms," *International Journal on Artificial Intelligence Tools*, vol. 20, no. 4, pp. 663–690, 2011.

[11] E. Argente, V. Julian, and V. Botti, "MAS Modeling based on Organizations," in *Post-Proceedings 9th International Workshop AOSE'08*, vol. 5386. Springer, 2009, pp. 16–30.

[12] E. Argente, V. Botti, and V. Julian, "Gormas: An organizational-oriented methodological guideline for open mas," in *10th Int. Workshop on Agent-Oriented Software Engineering (AOSE)*, 2009, pp. 85–96.

[13] S. Kelly, K. Lyytinen, and M. Rossi, "Metaedit+ a fully configurable multi-user and multi-tool case and came environment," in *Advanced Information Systems Engineering*. Springer, 1996, pp. 1–21.

[14] F. Zambonelli, N. Jennings, and M. Wooldridge, "Developing multiagent systems: The Gaia methodology," *ACM Transactions on Software Engineering and Methodology (TOSEM)*, vol. 12, no. 3, p. 370, 2003.

[15] S. DeLoach, *Multi-Agent Systems: Semantics and Dynamics of Organizational Models*. IGI Global, 2009, ch. Organizational Model for Adaptive Complex Systems, pp. 1–26.

[16] A. Molesini, E. Denti, and A. Omicini, "From aose methodologies to mas infrastructures: The soda case study." Springer, 2008, pp. 300–317.

[17] R. Ricci, M. Viroli, and A. Omicini, "CArtAgO: An Infrastructure for Engineering Computational Environments," in *3rd Inter. Workshop Environments for Multi-Agent Systems*, 2006, pp. 102–119.

[18] R. Ali, V. Bryl, G. Cabri, M. Cossentino, F. Dalpiaz, P. Giorgini, A. Molesini, A. Omicini, M. Puviani, and V. Seidita, "MEnSA Project - Methodologies for the Engineering of complex Software systms: Agent-based approach," UniTn, Tech. Rep. 1.2, 2008.

[19] M. Cossentino, "From requirements to code with the passi methodology," *Agent Oriented Methodologies IV*, pp. 79–106, 2005.

[20] P. Bresciani, P. Girogini, F. Giunchiglia, J. Mylopoulos, and A. Perini, "Tropos: An agent-oriented software development methodology," *Autonomous Agent and Multi-Agent Systems*, vol. 8, pp. 203–236, 2004.

[21] M. Moreno-Luzón and F. Peris, "Strategic approaches, organizational design and quality management: Integration in a fit and contingency model," *International Journal of Quality Science*, vol. 3, no. 4, pp. 328–347, 1998.

[22] A. Giret, V. Julian, M. Rebollo, E. Argente, C. Carrascosa, and V. Botti, "An open architecture for service-oriented virtual organizations," in *Seventh international Workshop on Programming Multi-Agent Systems.PROMAS 2009*, 2009, pp. 74–88.