

Document downloaded from:

<http://hdl.handle.net/10251/40667>

This paper must be cited as:

Esparcia García, S.; Argente Villaplana, E.; Centeno, R.; Hermoso, R. (2011). Enhancing MAS environments with organizational mechanisms. *International Journal on Artificial Intelligence Tools*. 20(4):663-690. doi:10.1142/S0218213011000395.



The final publication is available at

<http://dx.doi.org/10.1142/S0218213011000395>

Copyright World Scientific Publishing

International Journal on Artificial Intelligence Tools
© World Scientific Publishing Company

ENHANCING MAS ENVIRONMENTS WITH ORGANIZATIONAL MECHANISMS

SERGIO ESPARCIA, ESTEFANÍA ARGENTE

*Grupo de Tecnología Informática - Inteligencia Artificial
Departamento de Sistemas Informáticos y Computación
Universitat Politècnica de València
Camino de Vera, s/n - 46022 Valencia, Spain
{sesparcia, eargente}@dsic.upv.es*

ROBERTO CENTENO

*Departamento de Lenguajes y Sistemas Informáticos
Universidad Nacional de Educación a Distancia
C/ Juan del Rosal, 16 - 28040 Madrid, Spain
rcenteno@lsi.uned.es*

RAMÓN HERMOSO

*Centre for Intelligent Information Technologies (CETINIA)
University Rey Juan Carlos
C/ Tulipán, s/n - 28933 Móstoles (Madrid), Spain
ramon.hermoso@urjc.es*

Received (Day Month Year)

Revised (Day Month Year)

Accepted (Day Month Year)

This work proposes a new coordination system for the environment of a Multi-Agent System by merging the features from two important contributions to this field of research, Organizational Mechanisms and Artifacts. Organizational mechanisms can be introduced into a Multi-Agent System with the aim of influencing the behavior of agents populating it to achieve their goals in a proper way. In this paper, we propose to model organizational mechanisms by means of artifacts, which are non-proactive entities used by agents. Artifacts were presented within the Agents & Artifacts conceptual framework, and that present good advantages for coordinating agents' environments. We put forward a formal model that defines how organizational mechanisms can be designed by using artifacts theory. We validate the approach by presenting a case study focused on a real health care domain problem. Additionally, the Artifacts for Organizational Mechanisms are compared with some different proposed artifacts.

Keywords: Artifacts; Organizational mechanisms; Environment; Agent-Oriented Software Engineering

2 *S. Esparcia, R. Centeno, R. Hermoso, E. Argente*

1. Introduction

There are different approaches to develop Multi-Agent Systems (MAS), ranging from closed, agent-centered systems to open, organization-oriented systems.¹ When developing the latter of systems, known as Organization Centered MAS (OCMAS²), it can be noticed that the environment surrounding MAS is mainly considered as heterogeneous, unpredictable, distributed and dynamic.³ Being such a complex environment, it must include mechanisms and tools that help managing and controlling it.

Nowadays, the environment of a MAS⁴ is being modeled as a first class abstraction of the system. Different approaches presented new concepts that help developers to model the environment. One of the most recognized is the **Agents & Artifacts** (A&A) conceptual framework,⁵ which is based on the human cooperative elements and it is characterized by three types of abstractions: (i) *agents*, the proactive elements of the system; (ii) *artifacts*, the entities that must be used by the agents; and (iii) *workspaces*, a portion of the environment that contains agents and artifacts and defines the topology of the system. Additionally, the workspace is the space where agents and artifacts are able to develop their functionality.

Organizational mechanisms⁶ can be a valid method to provide coordination into organizations. They are mechanisms introduced in a MAS with the aim of influencing the agents' behavior towards more effectiveness with regard to some goals from both a macro and a micro perspective. Hence, these mechanisms can provide additional information to agents which may persuade them to behave in a certain way; or they can produce changes in the environment that may impose certain behaviors to agents. Thus, it is very useful to use these mechanisms in an open system where external agents are located, so then being able to promote coordination.

Seeing that artifacts are located into the MAS environment and they can also improve coordination between agents, the objective of this work is to model organizational mechanisms as artifacts, in order to facilitate system designers its usage and implementation. A generic idea of every mechanism will be given, so that MAS developers will be allowed to create the most effective artifact for their system. We will only define the minimum features, properties and operations, that the artifacts must provide to be considered as artifacts for organizational mechanisms. Additionally, a case study of application of this type of artifacts will be given, based on a real problem from the health care domain.

The rest of this paper is structured as follows: Section 2 describes the background of this work, by defining organizational mechanisms and artifacts. Section 3 models the organizational mechanisms as artifacts. Section 4 presents a case study based on the health care domain. Section 5 compares some of the existing artifacts with our proposal. Finally, Section 6 gives our conclusions on this proposal.

2. Background

This section describes the two paradigms our proposal is based on. On the one hand, organizational mechanisms will be described to give an overview on how they can improve the behavior of the agents in a MAS. On the other hand, artifacts will be depicted to show that they are a tool that facilitates the interaction between agents and their environment.

2.1. Organizational mechanisms

Organizational Mechanisms⁶ are mechanisms introduced in a multi-agent system with the aim of influencing the agents' behavior towards more effectiveness with regard to the global purpose of the system. They rely on the assumption that agents participating in the system are rational, i.e. try to maximize their utility with any action they perform. In order to clarify formalization we first give a definition of MAS on which the remainder formulae rely. We adhere the definition given by Centeno *et al.*⁶

Definition 1. A Multi-Agent System (MAS) is a tuple $\langle Ag, \mathcal{A}, \mathcal{X}, \Phi, x_0, \varphi \rangle$ where:

- Ag is a set of agents; $|Ag|$ denotes the number of agents in the system;
- \mathcal{A} is a finite action space that includes all possible actions that can be performed in the system. \mathcal{A} includes an action a_{skip} , the action of doing nothing;
- \mathcal{X} is the environmental state space;
- $\Phi : \mathcal{X} \times \mathcal{A}^{|Ag|} \times \mathcal{X} \rightarrow [0..1]$ is the MAS transition probability distribution, describing how the environment evolves as a result of agents' actions;
- $x_0 \in \mathcal{X}$ stands for the initial state of the MAS;
- $\varphi : Ag \times \mathcal{X} \times \mathcal{A} \rightarrow \{0, 1\}$ is the agents' capability function describing the actions agents are able to perform in a given state of the environment. $\varphi(a, x, ac) = 1$ ($\varphi(a, x, ac) = 0$) means that agent a is able (not able) to perform action ac in the state x .

Upon this definition of MAS, two different types of organizational mechanisms may be defined: *informative* and *regulative* (that has two different types, incentive and coercive). Relationships between these mechanisms are depicted in Fig. 1.

Informative organizational mechanisms⁶ (from now on informative mechanisms) are defined as a function that given a partial description of an internal state of an agent and taking into account the partial view that the mechanism has of the current environmental state, it provides information, which may consist of a set of actions an agent can take but it is possibly not aware of, a recommendation of a particular action which is eventually a "good action" for the agent, or information about the consequences that a given action may have. Formally it is defined as follows:

$$\Gamma : \mathcal{S}' \times \mathcal{X}' \rightarrow \mathcal{I} \quad (1)$$

4 *S. Esparcia, R. Centeno, R. Hermoso, E. Argente*

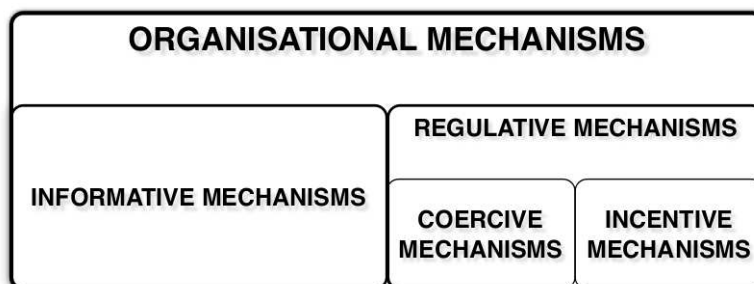


Fig. 1. *Types of Organizational Mechanisms*

where:

- \mathcal{S}' represents the set of possible partial descriptions of agents' internal states;
- \mathcal{X}' is the set of partial views of environmental states;
- \mathcal{I} represents an information space.

All informative mechanisms have in common that their usage is not imposed. Agents are free to use such mechanisms at their own discretion. To use an informative mechanism, an agent should provide it with part of his internal state. In fact, when rationality of agents is assumed,⁷ agents must use a given informative mechanism if and only if they expect that the usage of the mechanism will be advantageous for them. Informative mechanisms may improve the performance of individual agents and may have effects on the global performance of an organized MAS with respect to a global utility function. The information provided by this type of mechanisms will improve the knowledge of an agent, since the latter includes some extra information for reasoning and thus making him to better choose his future actions.

Regulative organizational mechanisms⁶ (from now on regulative mechanisms) share the same objective as informative mechanisms, but they focus on introducing changes into the environment in order to keep agents from undesired behaviors that drive the system to non-profitable states, that is, these mechanisms are in charge of producing changes in the system so as to reach states that improve the system's global utility. The rationale behind introducing changes in the environment is that agents perceive those changes, so possibly altering their reasoning to decide which action perform next. Such type of mechanisms rely on the existence of a system designer, which defines the preference relation over system states represented through the global utility function, and that has sufficient authority to impose certain changes in the system.

Two types of possible changes in the environment are considered: (i) introduction of incentives in order to make agents follow a desired behavior, and (ii) changes in the agents' action space. Accordingly, two types of regulative mechanisms have been

defined:

- An **incentive mechanism** is a function that given a partial description of an environmental state of MAS produces changes in the transition probability distribution of MAS. Formally:

$$\Upsilon_{inc} : \mathcal{X}' \rightarrow [\mathcal{X} \times \mathcal{A}^{|\mathcal{A}g|} \times \mathcal{X} \rightarrow [0..1]] \quad (2)$$

- A **coercive mechanism**, Υ_{coe} , for *MAS* is a function that given a possibly partial description of an environmental state of *MAS* produces changes in the agents' capability function of *MAS*, thus adding or deleting actions from an agent's action space.

$$\Upsilon_{coe} : \mathcal{X}' \rightarrow [Ag \times \mathcal{X} \times \mathcal{A} \rightarrow \{0,1\}] \quad (3)$$

where \mathcal{X}' represents the set of possible partial descriptions of the environmental states of *MAS*.

- Any regulative mechanism is either incentive or coercive.

Incentive mechanisms may produce changes in the consequences of agents' actions by introducing rewards and penalties. Obviously, rewards and penalties may produce variations in the expected utility of an agent's actions and, hence, rational agents would change their decisions accordingly (if they know about such incentives). Therefore, agents must be informed about the rules governing the system. In the case of coercive mechanisms the changes in the system are produced through a modification of agents' action space. New actions may be added or existing actions may be eliminated.

Both types of mechanisms emerge as an important contribution to MAS. Since nowadays MAS have progressively become open and heterogeneous, and it is possible that non-collaborative agents populate a system, it is necessary to endow them with mechanisms that help the system's administrators to keep the MAS under control. Both informative and regulative mechanisms are very useful to afford this task. Aside from this, organizational mechanisms need to be implemented into the environment of a MAS. The information provided by the organizational mechanisms can be supplied to the agents by using artifacts, being an important contribution to improve the MAS environment.

2.2. Artifacts

Artifacts⁵ are non-proactive, but reactive entities that agents employ to achieve their goals. As artifacts do not have assigned goals, they are associated to the goals of the agent that uses the artifact. To accomplish these goals, artifacts provide a *functionality*, which is partitioned into some *operations* that agents can execute when interacting with them. These operations are part of the *usage interface* of the artifact, which is completed with the *observable properties* that agents can check without invoking any operation in it. Artifacts provide a second group of operations,

6 *S. Esparcia, R. Centeno, R. Hermoso, E. Argente*

called *link operations* (accessible through a link interface) that enables composition of artifacts and load distribution, since different artifacts may be located at the same or different workspaces,⁵ which is the portion of environment that is perceived by an agent, who is able to interact with. Every workspace contains a set of artifacts; and the set of workspaces composing the environment is used to define its topology. Finally, artifacts are enhanced with a *function description* (which acts as a manual) and a set of *operating instructions*, an essential feature when dealing with open systems, since external agents can discover artifacts and evaluate whether they could be useful to reach their goals.

Since artifacts are very malleable components from the environment of a MAS, designers can develop new types of artifacts according to actual system needs. The Agent Oriented Software Engineering (AOSE⁸) community has already developed different types of artifacts⁹: (i) *basic artifacts*, which comprises artifacts that give information of very general world features (for example, clocks, calendars and timetables); (ii) *coordination artifacts*,¹⁰ which improve the coordination between agents in a MAS; (iii) *reputation artifacts*¹¹ that manage reputation values of agents in an organization; (iv) *cognitive stigmergy artifacts*,¹² which provide information about an agent or a society of agents that can be useful to other agents or groups; (v) *organizational artifacts*,¹³ which are used to manage an organization; and (vi) *argumentation artifacts*,¹⁴ which manage arguments between agents. Moreover, it is possible to use the *CArtAgO* framework¹⁵ to implement artifacts, which is engineered upon the principles of the A&A conceptual framework.⁵

3. Artifacts and the Organizational Mechanisms

This section describes how both types of organizational mechanisms, informative and regulative, can be modeled as artifacts. Artifacts allow an easy merging of the organizational mechanisms into the environment of a MAS.

We formalize an artifact as follows:

Definition 2. An Artifact is a tuple $\langle PR, OP, LO, St \rangle$ where:

- *PR* are the observable properties of the artifact that agents can directly check without operation invoking;
- *OP* is the set of operations that agents can execute when interacting with it;
- *LO* stands for link operations, which can be called by other artifacts. This type of operations enables artifact composition and functionality distribution by linking artifacts. In some cases, these operations may be used to help the initialization of another artifact;
- *St* is the internal state of an artifact, which is not accessible by the agents populating the system.

The result of this modeling is a set of three types of artifacts. The *informative artifacts* are based on the informative mechanisms; the *incentive artifacts* are based

on the incentive mechanisms; while the *coercive artifacts* are based on the coercive mechanisms.

3.1. Informative artifacts

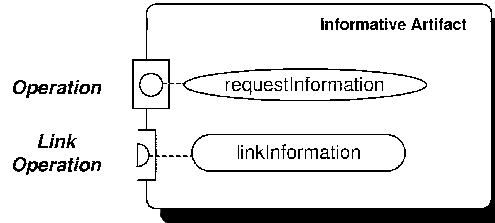
As stated previously, informative mechanisms return information about actions to an agent, given a partial description of his internal state and taking into account the partial view of the environment that the mechanism has. The informative mechanism has been modeled as an artifact, named *informative artifact*, being a passive entity that is used by agents in order to help them in their deliberative process.

Definition 3. An Informative Artifact is defined as an artifact $Ar_{inf} = \langle PR, OP, LO, St \rangle$ where:

- $PR \subseteq \{St \cup \emptyset\}$ are the observable properties of the informative artifact, which are a subset of the information contained into the artifact or an empty set.
- $OP : \mathcal{S}' \rightarrow \mathcal{I}$ are the operations of the artifact, where:
 - \mathcal{S}' represents a partial description of an agent's internal state.
 - \mathcal{I} represents the information returned by the artifact, based on the internal state of the artifact and the partial description of the agent's internal state (semantically, $\mathcal{S}' \times St \rightarrow \mathcal{I}$).
- $LO : \Theta \rightarrow \mathcal{I}$ is a link operation that is used by an artifact Ar_1 to obtain information from the Ar_{inf} artifact, where:
 - $\Theta \subseteq (\Sigma \cup \mathcal{S}')$ is the information sent by Ar_1 to Ar_{inf} ;
 - $\Sigma \subseteq \{St_1 \cup \emptyset\}$ is a partial state of Ar_1 , being St_1 the internal information of Ar_1 ;
 - \mathcal{S}' represents a partial description of the internal state of the agent that is requesting information to the artifact Ar_1 ;
 - \mathcal{I} represents the information returned by the artifact Ar_{inf} to the artifact Ar_1 (previously requested), based on the partial description of Ar_1 (Σ), the partial description of the agent's internal state who is requesting Ar_1 (\mathcal{S}') and the internal state of the artifact Ar_{inf} (St). Semantically: $(\Sigma \cup \mathcal{S}') \times St \rightarrow \mathcal{I}$.
- St represents the internal state of the artifact, i.e. the information contained into the artifact, which is not directly accessible by agents or other artifacts.

Informative artifacts are not required to provide with link operations, so they might be only accessible by agents in their same workspace. When they offer a link operation, artifacts located in their same workspace or in any other connected workspace can obtain relevant information from this informative artifact by means of its link operations.

Fig. 2 shows a graphic representation of an informative artifact. As explained before, this type of artifact needs, at least, one operation: `requestInformation(St)`.

8 *S. Esparcia, R. Centeno, R. Hermoso, E. Argente*

 Fig. 2. *Informative artifact for an informative mechanism*

Giving a partial description of the internal state of the agent, which can contain the roles, beliefs, facts and other features associated with the agent and his environment, this operation returns a package of information that contains: (i) the type of the information, which can be a recommendation or an advice about actions, or information about the consequences of executing an action; (ii) the description of this information, and (iii) a set of actions that are related to this information (they could be services that an agent could take, recommended services, services that have consequences to the agent, etc.). Moreover, as explained before, an informative artifact could need to make requests to other artifacts in order to obtain information or update its internal state. For this reason, the informative artifacts are enhanced with a link operation, $\text{linkInformation}(S', St)$, which is executed as a consequence of an operation (OP) invoked in another artifact. The requester artifact, Ar_1 , sends the partial state of the agent (agent's internal state) that requested information, as well as a subset of its internal information. As a result, the requested artifact, Ar_{inf} , will return some piece of information also based on its own internal state.

To exemplify how this type of artifacts work, we define an artifact that publicly provides norms currently active in the system. This artifact would contain operations shown in Fig. 3.

Let $Ar_{inf}^{norms} = \langle \emptyset, \{requestNorms\}, \{linkInfo\} \rangle$ be an artifact that aims to provide agents (on demand) with information about norms, such as the specification of norms that rule a role, active or deactivated norms, etc. The operation $linkInfo$ may be used by other artifacts in order to gather information related to norms that could improve their usage. Thus, the artifact encapsulates functionality for both, agents that request information for their personal purposes, and other artifacts that could also be interested in some information that the artifact manages about norms.

We can observe that there does not exist any observable property, since norms cannot be directly accessed by agents, but they may be requested by using the operation $requestNorms$. Consider that, since it is an informative artifact, the agent requesting for norms must send a part of his mental state in order to allow the artifact to give him back some useful information.

Notice that this artifact is not a mere repository of norms, since allows to be tuned to distinguish among different types of information that should be provided to agents. Thus, the mechanism designer probably does not want that any agent

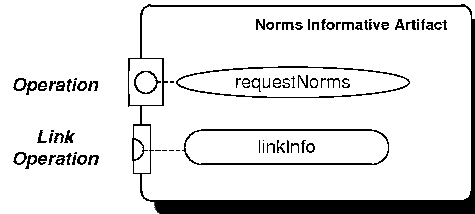


Fig. 3. Informative artifact for informing about norms

could know all the norms at any time, but it could probably prefer to give the precise information the agent is interested in, in such a way that it does not disclose any sensitive information.

A typical scenario would consist on an agent requesting for the set of norms that rule a specific role that the agent wants to play. Responsibilities, duties and rights that roles specify for its enactment should make the artifact to provide suitable information on demand.

3.2. Incentive artifacts

As explained before, incentive mechanisms are mechanisms that are able to produce changes in the system environment from a global view, modifying the rewards and penalties that are active in the system. These mechanisms rely on the belief that a possibly little change in the incentive system (that can be affecting only to a small number of agents) affects the entire system. In this subsection, the incentive mechanisms are modeled as artifacts, named *incentive artifacts*. These artifacts will execute organizational changes, which bring the possibility of implementing an adaptive system, by varying elements from the system (e.g. adding or deleting norms). After a change in the incentive system of the MAS is produced, transition probabilities between different states of the system are affected. In order to carry out these changes, it is necessary to have an agent or a human playing a special role that we call "system adapter", which is able to manage organizational changes when necessary to promote the adaptiveness of the MAS. The system adapter is the only agent that has privileges to execute the operations of an incentive artifact.

Definition 4. An Incentive Artifact is defined as an artifact $Ar_{inc} = \langle PR, OP, LO, St \rangle$ where:

- $PR \subseteq \{St \cup \emptyset\}$ are its observable properties;
- $OP : \Delta$ is the operation that allows the *system adapter* to introduce or remove incentives in the system;
- $LO = \emptyset$, since this type of artifacts has no predefined link operations;
- St represents the internal state of the artifact.

The operation of the artifact (OP) modifies the transition probability between different states of the system. This operation is defined as:

10 *S. Esparcia, R. Centeno, R. Hermoso, E. Argente*

$\Phi = St \rightarrow [\mathcal{X} \times \mathcal{A}^{|\mathcal{A}g|} \times \mathcal{X} \rightarrow [0..1]]$, where:

- Φ is the MAS transition probability distribution, describing how the environment evolves as a result of agents' actions.
- \mathcal{X} is the environmental state space.
- $\mathcal{A}^{|\mathcal{A}g|}$ is the set of actions executed by agents between two states of the MAS.

This operation works as follows: the agent provides some piece of information to the artifact, which might change its internal state (St). Given this new internal state, the transition probability between two states of the system is modified, so the behavior of the MAS changes in a global perspective.

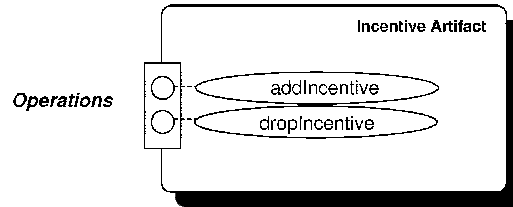


Fig. 4. *Incentive artifact*

Fig. 4 shows how an incentive artifact is modeled, including its minimum required features. The OP set contains two different operations: **addIncentive**($\{ty, ro, ac\}$), with $\{ty, ro, ac\} \in St$, which adds an incentive to the incentive system of the MAS; and **dropIncentive**($\{ty, ro, ac\}$), with $\{ty, ro, ac\} \in St$, which drops an incentive from the MAS. In these functions ty stands for the type of incentive (reward or penalty), ro refers to the role or set of roles affected by this incentive and ac represents a set of actions that are related with this incentive.

Sometimes, it could be useful to send information about changes in the incentive system to the agents populating the MAS. In order to execute this task it is necessary to provide the environment with an informative artifact, modeled as explained in the previous subsection.

To exemplify the incentive artifacts, we have chosen an organizational environment related to norms again.

Let $Ar_{inc}^{norms} = \langle \emptyset, \{addNormIncentive, dropNormIncentive\}, \emptyset \rangle$ be an incentive artifact that allows introducing positive incentives (*rewards*) and negative incentives (*penalties*) into an organization. These incentives consist of a set of possible consequences that norm fulfilment or violation, respectively, may entail. As aforementioned, the incentive mechanisms aim to improve the system performance by introducing changes in the environment that somehow influence the agents' reasoning. For this example we consider that the artifact does not contain any observable property and that it does not offer any minimum link operation to be requested by

other artifacts. The usage interface (OP) should not be available for every agent participating in the system. That is, this kind of artifacts does not provide information, but changes the environment, so only agents with sufficient permissions to do it should use operations in OP , depending on the domain. In our case agents capable of playing role "system adapter" can employ $addNormIncentive$ operation, so then attaching a penalty to a norm in case of violation; or introducing rewards for norm fulfilments. Incentives may also be updated through the time, by using $dropIncentive$ operation to remove the former and then updating with the new one by using $addNormIncentive$ operation.

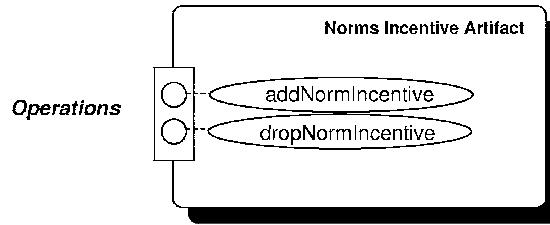


Fig. 5. Incentive artifact for norms

3.3. Coercive artifacts

As explained before, coercive mechanisms are aimed to produce changes in the environment of the system by producing changes in the agents' capability functions, given a possibly partial description of MAS. As it occurs with incentive mechanisms, coercive mechanisms are also relying on the existence of the "system adapter" role, which is able to promote organizational changes.

Formally, a coercive artifact is defined as:

Definition 5. A Coercive Artifact is an artifact $Ar_{coe} = \langle PR, OP, LO, St \rangle$ where:

- $PR \subseteq \{St \cup \emptyset\}$ are its observable properties;
- $OP : St \rightarrow [Ag \times \mathcal{X} \times \mathcal{A} \rightarrow \{0, 1\}]$ is the operation carried out by the coercive artifact, where:
 - Ag is an agent of the MAS;
 - \mathcal{A} is the action space that includes all possible actions that can be performed in the system.
- $LO = \emptyset$, since this type of artifacts has no predefined link operations;
- St represents the internal state of the artifact.

The operation $St \rightarrow [Ag \times \mathcal{X} \times \mathcal{A} \rightarrow \{0, 1\}]$, given the artifact's internal state, returns the capability for executing an action or not, 1 and 0 respectively. Internally, this operation works as follows: the artifact needs its internal state (St) as well as

the information provided by the system adapter (\mathcal{Ag} and \mathcal{A}) in order to execute this operation. After compiling all this information, the artifact calculates the new action space of the agent. This change can be seen as a local change but, since agents are related between them, changes in a single agent might produce changes in a set of agents, i.e. in the global state of the MAS.

Similarly to the informative and incentive artifacts, the coercive artifacts do not have mandatory observable properties. In case of having them, they are a subset of the internal knowledge of the artifact. The number of the available observable properties will depend on the purpose of each artifact. Additionally, they are not required to have link operations.

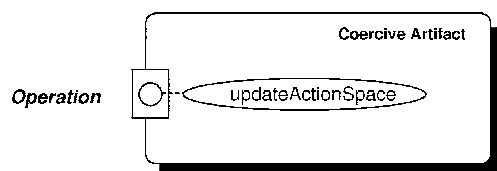


Fig. 6. *Coercive artifact*

Fig. 6 shows how a coercive artifact is modeled, including their minimum requirements. The only operation defined in this artifact is $\text{updateActionSpace}(\{ag, a\})$, where $\{ag, a\} \in St$ and $ag \in \mathcal{Ag}, a \in \mathcal{A}$, which receives an agent and an action from the system adapter and returns the capability for the given agent to perform this action.

As done with incentive artifacts, to show an example of coercive artifacts an organizational environment related to norms is taken:

Let $Ar_{coe}^{norms} = \langle \emptyset, \{updateActionSpace\}, \emptyset \rangle$ be a coercive artifact that aims to update agents' action spaces through time. As we stated in subsection 2.1, coercive mechanisms directly modify agents' action spaces to keep the former from undesirable behaviors. Thus, this artifact will be in charge of modifying those action spaces on demand of some special agents that have the permission to introduce these changes in the environment. Therefore, if one of the agents with sufficient permissions (i.e. "system adapter") observes that, for instance, the violation of a norm occurred, he could take the decision of banning some actions to the agent that did not fulfil that norm, trying to avoid that behavior in the future. In the same way that the artifact may remove actions from an agents' action space, it might also add actions to it, if agent's behavior is being acceptable. For instance, the system could test participants with a trial period to ensure that they behave accordingly to system's objectives, allowing them to perform more and more actions progressively.

Some examples of mechanisms that could be designed as incentive or regulative artifacts are: *normative manager*, that is, encapsulating dynamic consequences that fulfilment or violation of norms may entail; or *traffic sanctions manager*, where different sanctions may be applied about driving rules, even introducing constraints

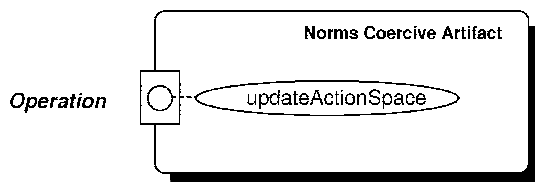


Fig. 7. Coercive artifact for norms

in the environment (roads can be closed, driver licenses could be taken away, etc.).

4. Case Study: Inter-Hospital Transfer Coordination

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.

Let us to show an example of this problem. Juan Domínguez, 23 years old, is waiting for assistance in the "Hospital de El Escorial",^a where he has been detected to have appendicitis. In this case, laparoscopic surgery is required. Juan would need to stay for another two days in 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 mean to provide the appropriate treatment in this hospital. Thus, Juan should be transferred to another hospital that has sufficient capacities to treat his case.

The administrative staff 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.

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.

Once the problem definition has been presented, we model different solutions

^aIt is a hospital located in the north of the region of Madrid

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 plan successfully 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 different organizational artifacts, aiming to fulfill the global objective.

4.1. *Solution based on argumentation (informative artifact)*

The first adopted solution relies on an informative artifact, $Ar_{inf}^{arg} = \langle \emptyset, \{newPatient, writeArguments, getArguments, getCommonSets\}, \emptyset \rangle$. In particular, such a mechanism encapsulates an argumentation algorithm able to propose a solution for a problem, based on arguments sent by different agents. Adapting it to our particular example, hospitals, represented in the system by autonomous agents, could argue about who of them would be in charge of attending a particular patient. They send their arguments, and the artifact calculates a solution over the shared arguments. Finally, the proposed solution is offered by the own artifact, and the coordination center by querying it, adopts such a solution.

As argumentation algorithm, we use the one presented by Oliva *et al.*¹⁴ The objective of that algorithm is to achieve a solution based on the arguments sent by agents. This process is divided in three steps: (i) agents share their arguments by sending them to the artifact (using operation `writeArguments`); (ii) the artifact executes the algorithm and calculates the conflict free and preferred extension over the shared arguments; and finally (iii) agents can query the artifact for getting the common sets calculated (operation `getCommonSets`). Besides, agents can also consult the artifact so as to understand and know other agents' arguments (by means of operation `getArguments`).

Therefore, in our example, when a new patient has to be transferred, the coordination center informs both, agents and artifact, about it, providing its particular characteristics and requirements. After that, agents send their arguments so as to (or not to) accept that patient. Then, the informative artifact calculates which agent (hospital) wins the argumentation process, in other words, which hospital has to admit such a patient. Fig. 8 depicts the informative artifact based on argumentation in charge of providing the solution.

Let us see a particular example showing how it works (see Fig. 9). Following the example introduced in section 4, "*Hospital de El Escorial*" calls the SUMMA112 coordination center to request the transfer of the patient called Juan. Then, all hospitals represented by their agents join the system with the global objective of assigning a destination to Juan. Let suppose that only two hospitals join the system: "*Hospital de Móstoles*" and "*Hospital Reina Sofia*" represented by *agent-1* and *agent-2*, respectively. Once they are in the system, the coordination center, represented by *agent-cc*, introduces in the informative artifact the record of the new patient `[Juan,23,"Hospital de El Escorial",appendicitis]`, so in-

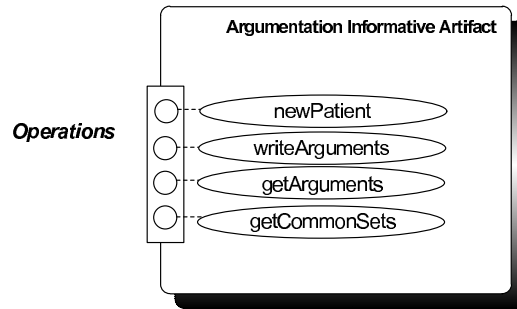


Fig. 8. Informative artifact encapsulating an argumentation algorithm

dicating name, age, origin hospital and diagnosis), by using the `newPatient` operation. After that, *agent-cc* asks hospitals for sending their arguments with regards to accepting the new patient. In this case, both agents representing hospitals send their arguments to the informative artifact; *agent-2* sends the argument [not enough bed capacities], while *agent-1* sends [carrying out clinical trial regarding appendicitis]. When both arguments are sent, the informative artifact calculates the result based on the argumentation algorithm and both hospitals and coordination center ask the informative artifact for the identifier of the patient and the artifact returns the destination hospital. In this example, the agent that clearly wins the argumentation is the *agent-1* due to it is interested in admitting the patient while *agent-2* is not. Therefore, the informative artifact returns [agent-1, "Hospital de Móstoles"] when it is queried with [patient, Juan]. Finally, the coordination center is able to plan the transfer preparing the necessary resources (ambulance, medical record, etc.).

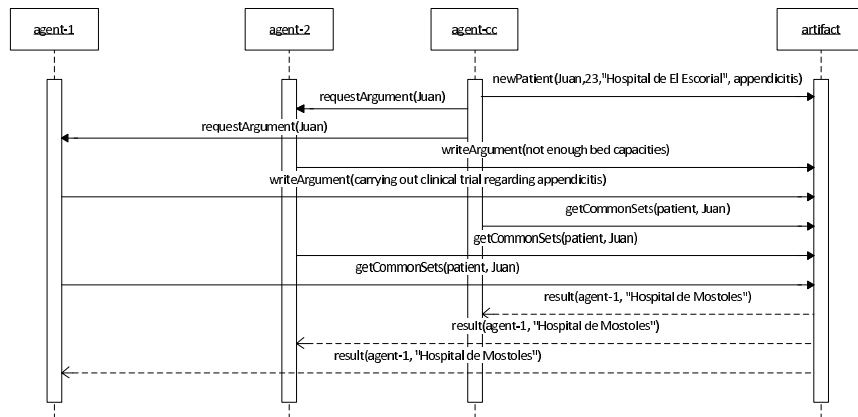


Fig. 9. Sequence diagram for the solution based on argumentation

4.2. *Solution based on an incentive artifact*

The solution presented in subsection 4.1, actually, is the solution adopted nowadays by the SUMMA112 coordination center, in the region of Madrid.^b However, such a solution suffers from some potential problems that, in fact, are currently happening. For instance, hospitals sometimes use false arguments so as to avoid to admit certain patients that could require a long or complicated treatment.

Dealing with these problems we propose another solution relying on an incentive artifact coupled with an informative one. The objective of the incentive artifact is to modify the consequence of some actions so as to agents have incentives to perform (or not) such actions. In this case, the coordination center is interested in agents to perform the action `admit a patient`. Thus, since the coordination center knows that hospitals are interested in having a high budget, the consequence of such an action will be modified such that the hospital's budget will be increased when it performs that action.^c Besides, in order to allow agents to reason about the new consequences, this information will be provided to agents by querying an informative artifact (see Fig. 10).

Therefore, when the coordination center receives a new request to transfer a patient, it joins the system, as well as all hospitals, with the aim of assigning a destination to such a patient. Since the coordination center (*agent-cc*) is the administrator of the system, it has enough permission to operate directly with the defined incentive artifact. Thus, when it joins the system it employs the operation `addIncentive(reward, hospital, admit patient)` (`dropIncentive(reward, hospital, admit patient)`) so as to add (drop) to the artifact the incentive that gives a *reward* to agents playing the role *hospital* when they perform the action *admit patient*. On the other hand, in order to allow agents to reason about the reward added to the system, there exists an informative artifact that provides the new consequences of that action. In this case, the artifact will provide the consequences (reward) of the action *admit patient*. This artifact, by means of a link operation with the incentive artifact (`addRewardInformation`), receives the consequences and the action it has to inform about. Therefore, agents are able to know about consequences of different actions before carrying them out.

Let us see how our example is solved now by using the proposed solution (see Fig. 11). Again, Juan needs to be transferred to other hospital. Thus, "*Hospital de El Escorial*" calls the SUMMA112 coordination center (*agent-cc*) and joins the system together with "*Hospital de Mostoles*" (*agent-1*) and "*Hospital Reina Sofia*" (*agent-2*). Once all of them are in the system, *agent-cc* employs the operation `addIncentive(budget+1000, hospital, admit patient)` provided by the

^bThe solution is not agent-based but human-based.

^cActually, lots of fluctuations in a hospital's budget do not make any sense, thus, the new consequences should be modified when a number of patients is admitted over a period of time, however, for the sake of simplicity and clarity, we assume that it is always modified when a hospital admits a patient.

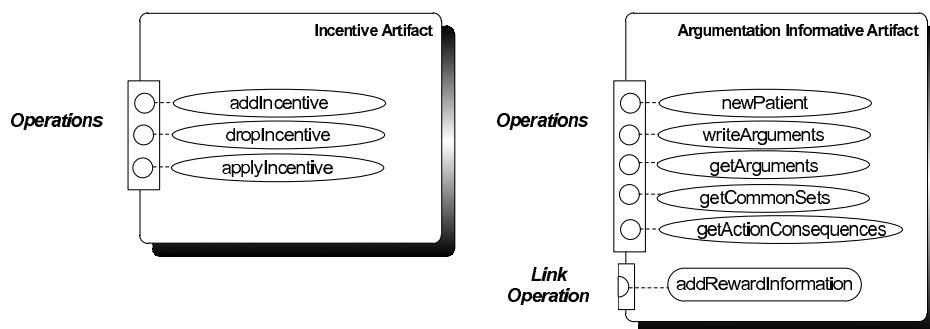


Fig. 10. Incentive and informative artifacts used in this solution

incentive artifact. At the same time, the incentive artifact employs the operation `addRewardInformation(admit patient, budget+1000)`, which is a link operation, over the informative artifact. Then, *agent-cc* informs hospitals, as well as the informative artifact, about the new patient, providing them with his information ([Juan, 23, "Hospital de El Escorial", appendicitis]). Once agents receive a message about a new patient, they have to deliberate on admitting it or not. Since agents already know about an informative artifact that informs about consequences of actions^d they query it (using operation `getActionConsequences`) providing two actions: `admit patient` and `reject patient`. In this case, let us suppose that *agent-2*'s argument about not having enough bed capacities was false, and now, knowing the new consequences for admitting a patient, it decides to admit Juan. Therefore, both hospitals are now interested in admitting Juan, and inform the *agent-cc* about it. So, the coordination center has to make a decision about which of both hospitals will be the destination for Juan. This decision could be taken based on some algorithm running in the own coordination center, or even the argumentation artifact presented in subsection 4.1 could be used, combining both solutions. Finally, once the *agent-cc* makes the decision, it informs hospitals about the patient's destination and it is the *agent-cc* itself who invokes the operation `applyIncentive` in the incentive artifact so as to give the winner the promised reward.

4.3. Solution based on a coercive artifact

So far we have put forward two different solutions for the inter-hospital transfer of patients, based on informative and incentive artifacts, respectively. However, there are still some cases in which outcomes resulting from the negotiation for the transfer

^dHow informative artifacts are discovered is out of the scope of this work. It could be done either by announcing them in a meta-informative artifact belonging to all systems, or because agents know about the existence of artifacts located at the same workspace as they are. It depends on the used metamodel.

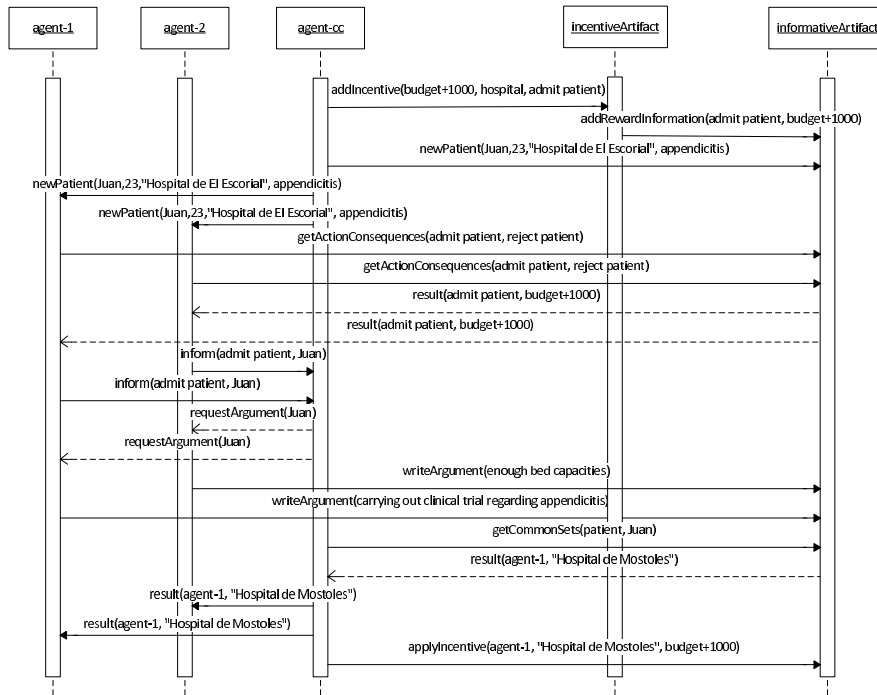


Fig. 11. *Sequence diagram for the solution based on an incentive artifact*

are not as they should be expected. The solution presented above attempted to incentive hospitals to admit patients by rewarding them for their positive attitude. Nevertheless, this could bring a drawback on the results. That solution fosters the positive behavior of hospitals but do not keep them from not accepting patients they are not interested in. A solution for avoiding this behavior could be to add an incentive artifact that tries to maximize the number of positive bids for accepting a patient when a transfer is needed. It would be designed in exactly the same manner than the one proposed in subsection 4.2 but imposing negative sanctions for not accepting patients, e.g. by reducing budget in some medical services. Even though, this solution entails some problems for any transfer negotiation process. In the end of any transfer negotiation process there is 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 which bade for the patient and those who did not. Even if the hospital really wants to accept the patient but it does not get it, would not be considered the same as if a hospital is not interested in the patient and does not bid for him/her. Some other non-desirable cases could happen, such as, for instance, a hospital that bids for a patient, but with so low level of interest. Therefore, this hospital that previously knows that the patient will not be assigned to it. This last case cannot be avoided

by using incentive artifacts.

For the reasons exposed above we point out another solution, based on the addition to the system of a coercive artifact (see Fig. 12) 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. Nonetheless, it is reasonable to think that a medical service cannot be removed suddenly just because the hospital that it belongs to rejected a patient. Thus, an intermediate solution would be to close a service only if the number of rejections of patients to be treated in it exceeds a given threshold.

As we have previously mentioned, the coordination center (*agent-cc*) acts as a system administrator, so it is supposed to have enough permissions to be able to modify the environment, that is, it has permissions to modify the artifact. Then, *agent-cc* will use the operation `updateActionSpace(constraint, hospital, admitPatient)`, which means that a new constraint is introduced into the system affecting agents playing role *hospital* regarding the action *admitPatient*. Moreover, as it occurred in the case of adding an incentive artifact, an informative artifact would be needed in order to show agents the consequences of not fulfilling new constraints. In our case, this artifact will provide the consequences of not admitting new patients. This artifact will contain a link operation that will be invoked by the coercive artifact so as to inform agents about new consequences about admitting a patient.

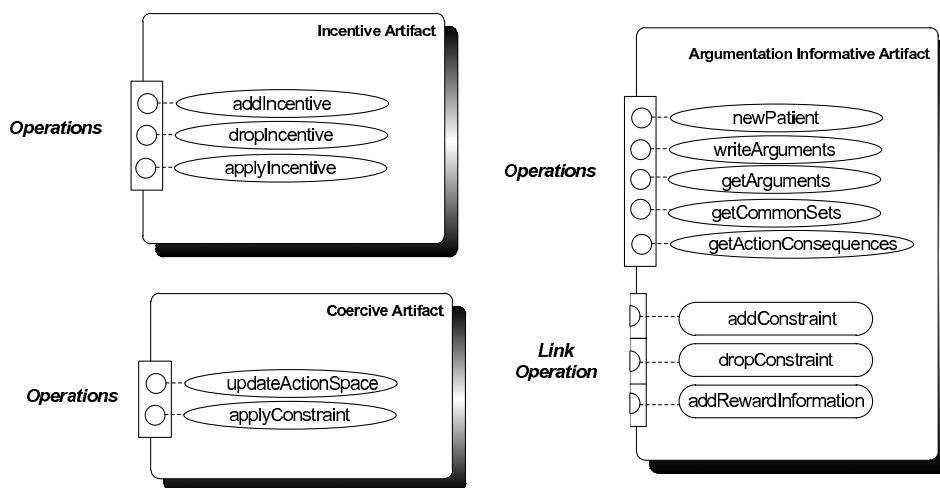


Fig. 12. Coercive, incentive and informative artifacts used in this solution

Following, we explain how this solution fits in our case study, as depicted in Fig. 13. "Hospital de El Escorial" needs to transfer Juan to other hospital. With this purpose it contacts SUMMA112 coordination center, represented by *agent-cc*, joining the system as well as "Hospital de Móstoles", represented by *agent-1*, and "Hospital Reina Sofia", represented by *agent-2*. Once this situation is reached, *agent-cc* uses the operation `updateActionSpace(closeMedicalService, hospital, ¬admitPatient)` of the coercive artifact explained above. Then, this artifact uses link operation `addConstraint(¬admitPatient, closeService)` of the informative artifact to show consequences of the action `¬admitPatient`. Even if the consequences do not specify the threshold of non-admitted patients for closing a service, the information about the possibility of closing a service should be enough to the hospitals for being concerned.

Then, as occurred in the other solutions the *agent-cc* informs hospitals about the new patient waiting for being transferred, providing them with this information ([Juan, 23, "Hospital de El Escorial", appendicitis]). Let us suppose that this coercive artifact is put in the environment together with the incentive and informative artifacts explained in the previous subsections. Then, agents representing hospitals will send queries to the informative artifact in order to know consequences of admitting or rejecting new patients (by means of `getActionConsequences`). Later, agents have to decide if their arguments, given in the first solution, are strong enough or they might be changed. If several hospitals bid for admitting the patient, the final choice of deciding the winner will correspond to the *agent-cc*, possibly based on the arguments given by an argumentation informative artifact (as proposed in the solution presented in subsection 4.1, if applied). After selecting the "winner" hospital, then *agent-cc* informs all participants and it invokes operation `applyConstraint(agent, ¬admitPatient)` for every hospital that rejected to accept Juan. Thus, the artifact reviews the rejections record of those agents that refused to admit the patient and decides if they should be penalized by closing one of their medical services, for instance, the one with less productivity.

Applying an incentive artifact as the one presented in Section 4.2 we do not avoid a non-collaborative behavior of the hospitals, but we just give rewards to those which are collaborative. With the solution proposed in current section we try to constraint that potential lack of collaboration of some hospitals, so trying to assure that when a hospital rejects an admission is because they cannot treat him/her properly.

5. Related Work

This section is aimed to compare our proposal with other available artifacts presented by different authors, in order to determine whether existing artifacts have the features of informative, incentive or coercive artifacts.

Informative artifacts provide a common functionality for MAS. Some of the arti-

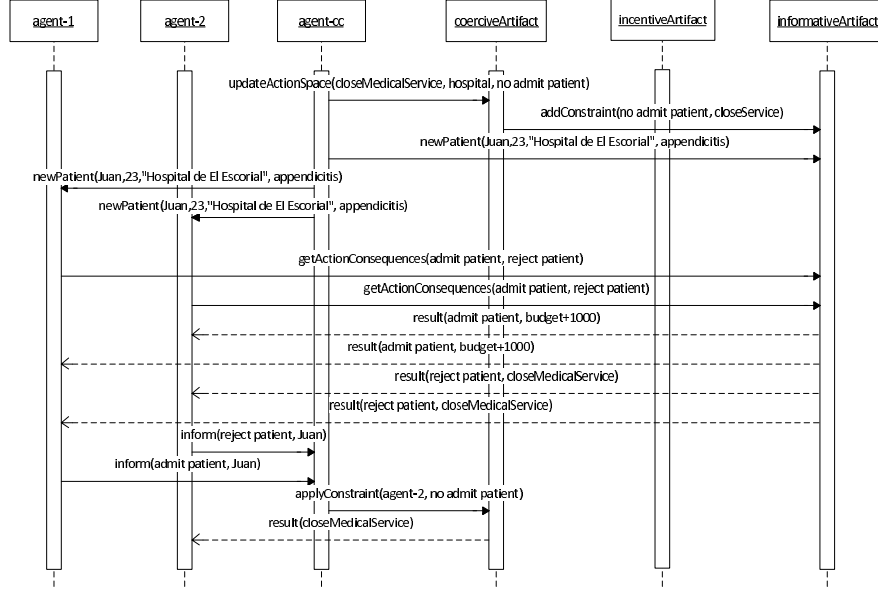


Fig. 13. Sequence diagram for the solution based on a coercive artifact

facts presented by the community of researchers provide information to agents after receiving information about a partial view of their internal state, so they could be seen as informative artifacts. For example, the *Role Evolution Coordination Artifact*¹⁶ that is aimed to build and evolve a role specialization taxonomy, which consists on a set of roles with a concrete order, over time; and make this information available to the agents. This artifact contains three operations: (i) *getBestRolesForInteraction*, which provides the most specialized roles for a given service type interaction; (ii) *getAgentsForRoles*, which provides the set of agents that play at least one of the roles in a given set of roles; and (iii) *getRolesForAgent*, which provides the set of roles that a given agent plays in the system.

Previously, in subsection 3.1, we have formally defined the operation of an informative artifact as $\mathcal{S}' \times \mathcal{St} \rightarrow \mathcal{I}$. A correspondence between the operations of the *Role Evolution Coordination Artifact* and the operation of an informative artifact can be established. In this way, the *getBestRolesForInteraction* operation function can be described as follows:

- $\mathcal{S}' = \text{Serv}$, where *Serv* is a service type interaction.
- $\mathcal{St} = R$, where *R* is the complete set of roles of the MAS.
- $\mathcal{I} = \mathcal{P}(R)$, where $\mathcal{P}(R)$ are the most specialized roles for *S*.

Similarly, the function *getAgentsForRoles* has the following correspondence:

- $\mathcal{S}' = \mathcal{P}(R)$, where $\mathcal{P}(R)$ is a set of roles.
- $\mathcal{St} = \text{Ag}$, where *Ag* is the complete set of agents of the MAS.

22 *S. Esparcia, R. Centeno, R. Hermoso, E. Argente*

- $\mathcal{I} = \mathcal{P}(\mathcal{A}g)$, where $\mathcal{P}(\mathcal{A}g)$ is the set of agents that play at least one of the roles in $\mathcal{P}(R)$.

Finally, the function *getRolesForAgent* presents the following correspondence with an operation of an informative artifact:

- $S' = \mathcal{A}g$, where $\mathcal{A}g$ is an agent of the system.
- $St = R$, where R is the complete set of roles of the system.
- $\mathcal{I} = \mathcal{P}(R)^e$, where $\mathcal{P}(R)$ is the set of roles that $\mathcal{A}g$ plays in the system.

Another example of artifacts that can be considered as informative artifacts are the argumentation artifacts,¹⁴ being the *Co-Argumentation Artifact* (CAA¹⁴) the most recognized one. This artifact gives assistance to argumentation processes. Participating agents share their arguments (i.e. a partial view of their internal state) with the artifact, which collect this information. Then, the artifact evaluates the arguments provided by all agents and calculates both the "social acceptability" (the acceptability of the arguments of a concrete agent) and the "social behavior" (the acceptability of the arguments from a global perspective). Using both values, agents take a final decision in their argumentative process, so their respective behaviors change. Then, it is also an informative artifact.

The CAA implementation provides two observable properties (*Social Behavior*, *Social Acceptability*) and one operation (*writeArguments*), which allows agents to store their arguments in the artifact.

Following the formalization of artifacts for organizational mechanisms, this CAA can be modeled as both an informative artifact and an incentive artifact. In this case, this artifact can be implemented with two different operations: *getSocialValues* and *writeArguments*.

It is possible to establish a correspondence between the operation *getSocialValues* of a Co-Argumentation Artifact and the required operation of an informative artifact. The partial description of an agent's internal state (S') is represented in a CAA as the argument that the agent will use during the argumentation process.

$$S' = Arg_t \quad (4)$$

where Arg_t is an argument provided by the agent t .

In this example, the internal state of the artifact (St) is the set of arguments that it has stored up to this moment.

$$St = \bigcup_{i=1}^n Arg_i \quad (5)$$

where St is the compilation of n arguments.

^e $\mathcal{P}(R)$ stands for the power set of R .

Finally, the information (\mathcal{I}) returned by the artifact are the values of the Social Acceptability and Social Behavior:

$$\mathcal{I} = \{SocAcc, SocBeh\} \quad (6)$$

where *SocAcc* is the Social Acceptability and *SocBeh* is the Social Behavior.

In a similar way, other types of artifacts, such as *Coordination artifacts*,¹⁰ or *Organizational artifacts*¹³ can also be described using artifacts for organizational mechanisms. Therefore, since coordination artifacts encapsulate a coordination service, this coordination service can be implemented by means of an informative artifact (providing useful information to the agents), an incentive artifact (modifying the transition probability between different states of the system) or a coercive artifact (allowing or banning agents from developing different actions). For example, the Follow Me coordination artifact¹⁰ is considered as an informative artifact, since its provided information is useful for agents in order to know their next action to execute (using operation *get*). Agents are also able to request the artifact to execute an action using the operation *do(a)*. Formally, operation *do(a)* from the Follow Me example can be described as:

- $S' = \emptyset$, since an agent do not need to provide information from his internal state.
- $St = \bigcup_{i=1}^n \mathcal{A}_i$, the set of available actions, where n is the total number of actions contained in the artifact.
- $\mathcal{I} \in \{true, false\}$ confirms if the action is executed or not.

The second operation from the Follow Me Artifact, *get* can be formally defined as:

- $S' = Ag$, being Ag the identifier of the agent that requests the operation.
- $St = \bigcup_{i=1}^n \mathcal{A}_i$, the set of available actions, where n is the total number of actions contained in the artifact.
- $\mathcal{I} = \mathcal{A}_i$ is the next action that the requester agent has to execute.

The observable property of this artifact is *act(a)*, that represents whether an action a is completed or not.

Regarding *organizational artifacts*, they are used to manage an agent organization in order to help the organization to reach its goals from a global, social level. A clear example of this type of artifacts is an artifact that helps informing or managing norms that, as it has been previously explained along Section 3, it can be modeled as an informative artifact (providing norms currently active in the system), an incentive artifact (introducing positive or negative incentives into an organization) or a coercive artifact (removing actions from agent's action space or including new possible actions, i.e. when an agent takes or leave a role).

The examples proposed in OR4MAS¹³ include three artifacts that can be identified with the different types of artifacts for organizational mechanisms. *OrgBoard* is an artifact that returns information about different aspects of the organization

such as the agents playing a role, so it is clearly an informative artifact. But, the link operation, used to add new elements inside an *OrgBoard* artifact, changes the system elements, so this link operation features properties from incentive artifacts. *GroupBoard* artifact is an incentive and coercive artifact because of their operations and it also has properties from informative artifacts, given by its link operation *isMember*, which given an agent returns whether an agent is a member of a group or not. This artifact is focused on the role enactment process of the system. Therefore, if an agent adopts or leaves a role he will be affected by a different reward system, according to the adopted/left role. Also, taking or leaving a role in different proposals such as Electronic Institutions (e-Institutions¹⁷) or THOMAS¹⁸ changes the available actions that an agent is able to perform, so it is a feature of a coercive mechanism. As an example, operation *AdoptRole*(ρ) from *GroupBoard* artifact, used by an agent to adopt a new role, being ρ the role to be adopted, is formalized using two types of operations, related to incentive and coercive artifacts. The operation related to incentive artifacts refers to the norms that affect the role being taken. Then, the reward system changes, thus modifying the probability of changing the system from one state to another. On the other hand, taking a role in the beforehand mentioned proposals also modifies the set of actions that an agent can perform, so this is a feature related to coercive artifacts. Formally, an operation from a coercive artifact is defined as $St \rightarrow [Ag \times \mathcal{X} \times \mathcal{A} \rightarrow \{0, 1\}]$.

In this particular case:

- St is the internal knowledge of the *GroupBoard* artifact.
- Ag is the agent taking the role ρ .
- \mathcal{X} is the current environmental state.
- \mathcal{A} is an action from the action's space.

Therefore, after executing the *adoptRole* operation, tuples containing the information about the agent's action space are added to the internal state of the artifact.

$$St' = St \bigcup (\forall ag \in Ag, \forall x \in \mathcal{X}, \forall a \in \mathcal{A} [ag, x, a \rightarrow \{0, 1\}]) \quad (7)$$

ORA4MAS's *SchemeBoard* is an incentive artifact, since it is focused on missions of the system. An agent is able to commit or leave a concrete mission, each of them is enhanced with a set of obligations, thus changing the reasoning of an agent. Finally, the *NormativeBoard* artifact presents features from informative and incentive artifacts. It is informative because provides information about norms. A norm is able to change the reward system that an agent is affected by, thus defining an incentive artifact.

We state that ORA4MAS is very similar to our proposal. Although both proposals can be considered as complementary, they are different. Both of them are focused on regulating and controlling the behavior of an Organization Centered MAS (OCMAS²) but our proposal has a more general point of view than ORA4MAS, which focuses on open normative multi-agent systems. Due to this nor-

mative approach, authors of ORA4MAS pay attention to mechanisms to instrument norms. They use two types of norm instrumentation mechanisms: regimentation and enforcement.¹⁹ Regimentation is a mechanism that avoid agents to perform an action that violates a norm. This mechanism is also used by a well-known proposal such as the e-Institutions approach. This operation can be carried out by a coercive artifact, which removes an action that can violate a norm from the action's space of an agent. On the other hand, ORA4MAS also applies enforcement, a reactive mechanism executed after the violation of a norm. Then, it decides the sanction or reward that should be applied. An enforcement mechanism can be implemented by using an incentive artifact that is able to modify the reward system of a MAS.

*Reputation artifacts*¹¹ encapsulate the collection of norm violations of the participants in a system and then aggregate them allowing agents to request information from a concrete agent, which is able to change the behavior of an agent. Therefore, a reputation artifact presents the properties of an informative artifact. For example, the reputation artifact¹¹ that helps Alice, a MSc. student, to select the best partner to write a paper with uses experiences previously collected and evaluated by the artifact, so Alice will be able to build a reputation for each partner based on the computed evaluation.

Artifacts for Cognitive Stigmergy¹² are implemented to promote awareness inside a system. Agents are aware of activities of other agents and some other different events inside the system using artifacts. These artifacts provide information about annotations made during the execution of a MAS. Therefore, Artifacts for Cognitive Stigmergy are considered as informative artifacts.

Finally, there are some basic artifacts, like a counter or a database,²⁰ that are considered to be informative artifacts.

Table 1 depicts the different types of artifacts identified⁹ related with the artifacts for organizational mechanisms that they can be designed with.

Relation between Artifacts and Organizational Mechanisms Artifacts	
Type of Artifact	Organizational Mechanisms
Coordination	Informative, incentive, coercive
Argumentative	Informative
Cognitive stigmergy	Informative
Organizational	Informative, incentive, coercive
Reputation	Informative

As shown in this section, features of existing artifacts can be modeled following the proposed formalization of artifacts for organizational mechanisms. In most cases, current operations offered by existing artifacts make them to be easily modeled as informative artifacts. Moreover, the proposed formalization can also be useful to extend operations of current artifacts so as to apply incentive and/or coercive mechanisms into the environment. In some cases, a concrete type of artifacts can be modeled in different ways. Moreover, an artifact can be also modeled as the combination of different types of artifacts for organizational mechanisms.

6. Conclusions

Organizational Mechanisms are aimed to improve coordination between agents in a MAS, trying to change this coordination from a micro perspective (i.e., the perspective of individual agents), providing useful information to the agents (informative mechanisms); and a macro perspective (i.e., the perspective of the whole MAS), by modifying either action consequences (incentive mechanisms) or agents' capability functions (coercive mechanisms).

In this work, these 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 explore all knowledge they have about entities populating the system.

As a future work, we are working on the integration of Artifacts for Organizational Mechanisms into an Organization Modeling Language,²¹ which is a metamodel²² that is conceived to develop Organization Centered MAS, the Virtual Organization Model (VOM²³). The addition of these artifacts will enhance the metamodel with new features that will improve the organizational capabilities of the agents populating the system. Additionally, these artifacts can be also implemented using CArtAgO.¹⁵ Using this framework, it is possible to define patterns for each type of artifacts. In our case, a library for artifacts for organizational mechanisms will be defined. Moreover, the presented artifacts will be implemented using THOMAS¹⁸ in order to evaluate their performance with this framework, which is oriented to develop Virtual Organizations.²⁴

Acknowledgments

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

References

1. E. Argente, A. Giret, S. Valero, V. Julian, and V. Botti, "Survey of mas methods and platforms focusing on organizational concepts," in *Frontiers in Artificial Intelligence and Applications*, vol. 113, pp. 309–316, IOS Press, 2004.
2. C. Lemaître and C. Excelente, "Multi-agent organization approach," in *Proceedings of II Iberoamerican Workshop on DAI and MAS*, 1998.
3. A. Omicini, "SODA: Societies and infrastructures in the analysis and design of agent-based systems," in *Agent-Oriented Software Engineering*, pp. 311–326, Springer, 2001.

4. H. V. D. Parunak and D. Weyns, "Special issue on environments for multi-agent systems," *Auton. Agents Multi-Agent Syst.*, vol. 14, pp. 1–4, February 2007.
5. A. Ricci, M. Viroli, and A. Omicini, "Give agents their artifacts: the A&A approach for engineering working environments in MAS," in *Proceedings AAMAS*, p. 150, 2007.
6. R. Centeno, H. Billhardt, R. Hermoso, and S. Ossowski, "Organising MAS: a formal model based on organisational mechanisms," in *Proc. SAC*, pp. 740–746, 2009.
7. A. Rao and M. Georgeff, "Modeling rational agents within a BDI-architecture," *Readings in agents*, pp. 317–328, 1997.
8. C. Bernon, M. Cossentino, and J. Pavón, "Agent-oriented software engineering," *The Knowledge Engineering Review*, vol. 20, no. 02, pp. 99–116, 2005.
9. S. Esparcia and E. Argente, "A functional taxonomy for artifacts," in *Proceedings HAIS*, 2010.
10. A. Omicini, A. Ricci, M. Viroli, C. Castelfranchi, and L. Tummolini, "Coordination artifacts: Environment-based coordination for intelligent agents," in *Proc. AAMAS*, pp. 286–293, 2004.
11. J. Hubner, O. Boissier, and L. Vercoouter, "Instrumenting multi-agent organisations with reputation artifacts," *Proceedings of Coordination, Organizations, Institutions and Norms (COIN@ AAAI)*, pp. 17–24, 2010.
12. A. Ricci, A. Omicini, M. Viroli, L. Gardelli, and E. Oliva, "Cognitive stigmergy: Towards a framework based on agents and artifacts," in *Environments for MultiAgent Systems III* (D. Weyns, H. V. D. Parunak, and F. Michel, eds.), vol. 4389 of *LNAI*, pp. 124–140, Springer, May 2007. 3rd International Workshop (E4MAS 2006), Hakodate, Japan, 8 May 2006. Selected Revised and Invited Papers.
13. J. Hubner, O. Boissier, R. Kitio, and A. Ricci, "Instrumenting multi-agent organisations with organisational artifacts and agents," *Autonomous Agents and Multi-Agent Systems*, vol. 20, no. 3, pp. 369–400, 2010.
14. E. Oliva, P. McBurney, and A. Omicini, "Co-argumentation artifact for agent societies," *LNCS*, vol. 4946, p. 31, 2008.
15. A. Ricci, M. Viroli, and A. Omicini, "CArtAgO: An Infrastructure for Engineering Computational Environments," in *Proceedings E4MAS*, pp. 102–119, 2006.
16. R. Hermoso, H. Billhardt, and S. Ossowski, "Role Evolution in Open Multi-Agent Systems as an Information Source for Trust," in *Proc. AAMAS*, 2010.
17. M. Esteva, J. Rodriguez-Aguilar, C. Sierra, P. Garcia, and J. Arcos, "On the formal specification of electronic institutions," *Agent mediated electronic commerce*, pp. 126–147, 2001.
18. 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*, pp. 74–88, 2009.
19. D. Grossi, H. Aldewereld, and F. Dignum, "Ubi lex, ibi poena: Designing norm enforcement in e-institutions," *Coordination, Organizations, Institutions, and Norms in Agent Systems II*, pp. 101–114, 2007.
20. A. Ricci and M. Piunti, "Designing and Programming Agents' Environments in Multi-agent Systems," in *Handouts of the Eleventh edition of the European Agent System Summer School*, pp. 3–31, 2009.
21. V. Dignum, J. Meyer, H. Weigand, and F. Dignum, "An organization-oriented model for agent societies," in *Proceedings of International Workshop on Regulated Agent-Based Social Systems: Theories and Applications*, 2002.
22. J. Van Gigch, *System design modeling and metamodeling*. Springer, 1991.
23. N. Criado, E. Argente, V. Julian, and V. Botti, "Designing virtual organizations," in *Proceedings PAAMS*, vol. 55 of *Advances in Soft Computing*, pp. 440–449, 2009.

28 *S. Esparcia, R. Centeno, R. Hermoso, E. Argente*

24. T. Norman, A. Preece, S. Chalmers, N. Jennings, M. Luck, V. Dang, T. Nguyen, V. Deora, J. Shao, W. Gray, *et al.*, “Agent-based formation of virtual organisations,” *Knowledge-Based Systems*, vol. 17, no. 2-4, pp. 103–111, 2004.