Role Evolution in Open Multi-Agent Systems as an Information Source for Trust*

Ramón Hermoso
Centre for Intelligent Information Technology
University Rey Juan Carlos
Madrid, Spain
ramon.hermoso@urjc.es

Holger Billhardt
Centre for Intelligent Information Technology
University Rey Juan Carlos
Madrid, Spain
holger.billhardt@urjc.es

Sascha Ossowski
Centre for Intelligent Information Technology
University Rey Juan Carlos
Madrid, Spain
sascha.ossowski@urjc.es

ABSTRACT

In Open Multi-Agent Systems (OMAS), deciding with whom to interact is a particularly difficult task for an agent, as repeated interactions with the same agents are scarce, and reputation mechanisms become increasingly unreliable. In this work, we present a coordination artifact which can be used by agents in an OMAS to take more informed decisions regarding partner selection, and thus to improve their individual utilities. This artifact monitors the interactions in the OMAS, evolves a role taxonomy, and assigns agents to roles based on their observed performance in different types of interactions. This information can be used by agents to better estimate the expected behaviour of potential counterparts in future interactions. We thus highlight the descriptive features of roles, providing expectations of the behaviour of agents in certain types of interactions, rather than their normative facets. We empirically show that the use of the artifact helps agents to select better partners for their interactions than selection processes based only on agents' own experience. This is especially significant for agents that are newcomers to the OMAS.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence - Multiagent Systems

General Terms

Algorithms

Keywords

Artificial societies, Evolution/Adaptation, Self-organisation

1. INTRODUCTION

In recent years, the interest in Open Multi-Agent Systems (OMAS) has notably increased as a field of study of distributed AI. Agents participating in OMAS can join and leave the system at their will and, what is more important, may have been developed by different designers and programmers. The latter issue entails a potential variability among individuals' behaviours.

One of the major problems in OMAS is the question of how to coordinate their dynamics. Much work has been carried out on studying how to prescribe and enforce behaviour of the individual agents in order to reach some predefined global goal. These techniques are usually applicable in systems with certain characteristics: there exists an explicit global goal or purpose, and some authority that can enforce the prescribed behaviour. Other types of OMAS based on a more societal structure, e.g. agent societies [9], often neither have an explicit goal nor there exists a clear authority. In such systems, agents are embedded in some environment in which they can interact with others in order to improve their individual utility. And the coordination goal there consists in obtaining a more efficient behaviour of the individuals. Coordination mechanisms that provide guidance to agents, rather than prescribing and enforcing certain behaviours, seem to be suitable for such systems.

In this paper we present a proposal for a coordination mechanism for OMAS with a societal structure where agents try to improve their individual utility. In order to do that, agents may interact with others in the sense that they will delegate certain tasks or use certain services or capacities other agents may provide. We assume that agents in these systems accumulate their experiences of past interactions and implement some kind of trust model that allows them to establish which agents are more appropriate as possible interaction partners in the future. Based on these trust models, our approach evolves role taxonomies and assigns agents to roles. Agents can then request this information and use it in their decision-making processes. In particular, they can use the assignments of roles to agents in order to improve their trust models, that is, in order to evaluate the expected behaviour or outcome when delegating tasks to or using services from others.

The basic idea behind our approach is that in a society of agents social relationships may evolve. Such relationships define the positions of agents in the society in terms of capacities and importance as seen from others. Such positions can be identified as roles. Thus, knowing the positions agents have in a society (e.g., which roles they play) may

*The present work has been partially funded by the Spanish Ministry of Education and Science under projects TIN2006-14360-C03-02 and TIN2009-13839-C03-02 and by the Spanish project "Agreement Technologies" (CONSOLIDER CSD2007-0022, INGENIO 2010)

Cite as: Role Evolution in Open Multi-Agent Systems as an Information Source for Trust, Ramón Hermoso, Holger Billhardt and Sascha Ossowski, *Proc. of 9th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2010)*, van der Hoek, Kaminka, Lespérance, Luck and Sen (eds.), May, 10–14, 2010, Toronto, Canada, pp. 217-224

help the individuals to find appropriate partners for their interactions.

The concept of role is often considered from a macro perspective describing the objectives, goals and also the constraints applied to players in an organizational context. However, we consider roles from another perspective: as representing expectations of behaviours. In particular, we consider roles from a micro perspective (that is, from the perspective of the agents), where the role other agents are playing in the system provides information about their expected capacities regarding certain interactions (e.g., the provisioning of certain services or tasks). In this sense, our notion of role is rather related to social roles. The proposed mechanism evolves role taxonomies over time and adapts itself to changes in the system, what is useful when dealing with open and dynamic nature of OMAS.

Trust models are used to calculate the expectations to select partner to interact with. Our approach uses the notion of trust in two ways: i) as a source of information for establishing role taxonomies, and ii) the created role taxonomies themselves can be used as an additional dimension to evaluate trust at the agent level.

Furthermore, we implemented our mechanism as a coordination artifact [15]. Coordination artifacts can be conceived as entities specialised to provide a coordination service in a MAS. They are *infrastructure* abstractions meant to improve coordination activities and have proved to be suitable for developing collaborative working environments.

The remainder of the paper is organised as follows. In Section 2 we settle several assumptions and define our notion of role and role specialization taxonomies. Section 3 describes our role evolution mechanism in detail. In Section 4 we present the implementation of the mechanism as a coordination artifact. Section 5 specifies how the artifact can be used combined with the agents' trust models and presents some empirical results. Finally, conclusions, as well as the related and future work are presented in Section 6.

2. DEFINITIONS AND ASSUMPTIONS

An OMAS is defined as a set of agents embedded in an open environment, where agents may join or leave the system at any time at their own will. Furthermore, agents may be developed by different designers, which entails possibly different goals and preferences. In this paper we consider OMAS with a societal structure, i.e., that have no explicit global goal other than maximizing the utility of the individual agents.

Due to the openness, the agents in the system are neither obliged to pursue the same goals nor to share the same preferences. However, we assume agents to be rational entities. Rationality implies that agents use the *Maximum Expected Utility* principle to decide their actions as follows. Let $\mathcal{X}$ be the state space of the environment. Let $u : \mathcal{X} \Rightarrow \mathbb{R}$ be an utility function for a rational agent, assigning a value for every possible environmental state; and let $d : \mathcal{X} \Rightarrow \mathcal{A}$ be the function modelling the agent's decision process, obtaining the next action to perform (amongst the possible ones) in a given environmental state. This function is defined as:

$$d(x) = argmax_{a\in\mathcal{A}} eu(a, x) = \\ argmax_{a\in\mathcal{A}} \sum_{x'\in\mathcal{X}} u(x') \cdot \bar{P}(x'|x, a)$$

where $eu(a, x)$ is the expected utility for performing action a in state x; $u(x')$ is the utility estimate of state x'; and $\bar{P}(x'|x, a)$ is the agent's estimate about how likely is the transition from state x to x' when performing action a. It should be noted that agents do usually not know the exact probability distributions over possible outcome states when taking a given action a in a given state x. They just estimate these probabilities based on their experience and possibly other additional information. So the proposed coordination mechanism provides such additional information and thus, aims at improving the probability estimates $\bar{P}(x'|x, a)$.

We assume that the agents may interact with other agents in the system, in order to achieve their goals and to improve their utility. Furthermore, we assume that certain interactions in the system are typical *client/provider* (or *requester/requestee*) interactions: interactions where one agent delegates a task or uses a service provided by another agent. For simplicity, our mechanism only takes into account such "service type" interactions focusing on the provider side, that is, the agents offering services or task executions.

Our approach relies on the notion of role. When dealing with MAS, this concept has mainly been used from a macro perspective: i) to define different positions, at design time, that agents may hold at execution time; and ii) to regulate different interactions within the environment, giving some functionality to them by prescribing rights and duties derived from their enactment. However, especially in OMAS, where participants can change or evolve their behaviours throughout the time, the prescriptive nature of roles is only one of its facets. From a micro-level point of view – for an individual – a role an agent plays generates expectations about its behaviour or its capacities regarding certain actions. In this sense, we claim that: *i)* roles could be used as a descriptive concept in the system, instead of an entity for prescriptive enforcement; and *ii)* the knowledge about the roles agents play can be considered as a micro-level information unit. This means that rational agents could use such knowledge in their reasoning processes.

Considering that we only are interested in "provider" roles, we define a role in an OMAS as follows:

Definition 1. Let O be an OMAS and $\mathcal{A}g$ the set of agents participating in it. $\mathcal{A}$ is the set of possible actions that the environment allows agents to carry out. Among those, we can distinguish a subset $\mathcal{S} = \{s_1, ..., s_n\} \subseteq \mathcal{A}$ representing the set of service type interactions. Let $\mathcal{R}$ be a set of role identifiers. Then, a role in an OMAS is defined as a pair $\langle r, \mathcal{E}\rangle$ where

- $r \in \mathcal{R}$ is the role identifier;
- $\mathcal{E} = \{s_j, ..., s_k\} \subseteq \mathcal{S}$, is a finite set of (service type) interactions.

The intended semantics of a role $\langle r, \mathcal{E}\rangle$ is that agents playing the role r are qualified providers of the interactions contained in $\mathcal{E}$ in the sense that they are "skilful" for providing the services. We assume that all agents could, in principle, provide any service, but they will be qualified only for some of them. Based on the definition of role, we define a role specialization taxonomy.

Definition 2. A role specialization taxonomy in an OMAS O is a tuple $\mathcal{RT} = (R, \rhd_r)$ consisting of a set R of roles in O and a partial ordering $\rhd_r$ on R, such that:

1. $\exists r_{root} = \langle r_r, \mathcal{E}_r \rangle \in R:$

$$\mathcal{E}_r = \mathcal{S} \wedge \forall r \in R : (r = r_{root} \vee r_{root} \rhd_r r)$$

2. $\forall \langle r_1, \mathcal{E}_{r_1} \rangle, \langle r_2, \mathcal{E}_{r_2} \rangle \in R : \langle r_1, \mathcal{E}_{r_1} \rangle \rhd_r \langle r_2, \mathcal{E}_{r_2} \rangle \Leftrightarrow$

$$\mathcal{E}_{r_2} \subseteq \mathcal{E}_{r_1} \quad \wedge$$

$$\forall s \in \mathcal{E}_{r_2} : \quad \frac{\sum_{a \in \mathcal{A}_g} \sum_{b \in ag(r_2)} \mathcal{U}_a(b,s)}{|ag(r_2)| \cdot |\mathcal{A}_g|} \quad > \quad \frac{\sum_{a \in \mathcal{A}_g} \sum_{b \in ag(r_1)} \mathcal{U}_a(b,s)}{|ag(r_1)| \cdot |\mathcal{A}_g|},$$

where $ag(r)$ denotes the set of agents playing role r and $\mathcal{U}_a(b,s)$ denotes the estimate of the utility agent a expects when using the service s from agent b.

A role specialization taxonomy structures the roles by establishing a specialization relation $\rhd_r$ regarding the skills of the providers playing those roles. There is a root role r_{root} that all agents will play by default. All other roles are specialization of the root role or of another already specialized role. A role r_2 is a specialization of another role r_1 iff. i) the set of service type interactions specified by r_2 is a subset of those specified by r_1 and ii) for these interactions, the agents playing role r_2 are expected to perform better on average than the agents playing r_1. Agents may play several roles and if an agent plays a role r then it also plays all roles for which r is a specialization. Using these definitions, roles describe the capacities of agents as providers of certain services. When an agent is looking for another agent from which it wants to use a service, it can follow the information provided by the role taxonomy to *trust* in those agents that play the most specialized role for providing that service.

3. EVOLVING ROLE TAXONOMIES FROM AGENT EXPECTATIONS

Our mechanism aims at identifying agents that have been proven in the society to be "good" providers in particular service type interactions, and then group them into roles. In order to do that we have to identify behavioural patterns of agents in the system. As we said before, we assume that the agents in an OMAS are endowed with a trust model which allows them to estimate the expected utility they may receive when using a service provided by others. We use these values as the input for our evolution algorithm[1].

3.1 Trust Spaces in OMAS

Let O be an OMAS. $\mathcal{A}_g$ denotes the set of agents in O and $\mathcal{S} = \{s_1, .., s_n\}$ the service type interactions that are available. Furthermore, let $t_{a_i \rightarrow \langle a_k, s_j \rangle} \in [0..1]$ denote agent a_i's trust in agent a_k being a good partner (provider) in the service type interaction s_j. The trust values stored by agents provide a means to represent the service providing capacities of agents in the system. Each agent a_k can be represented as a vector $\widehat{a_k} = (\bar{s}_1, \bar{s}_2, ..., \bar{s}_n)$ in the n-dimensional vector space formed by all service type interactions in $\mathcal{S}$. Each $\bar{s}_j$ is defined as the average trust the other agents in the system have regarding a_k's capacity providing the service s_j:

$$\bar{s}_j = \frac{\sum_{a_i \in \mathcal{A}_g} t_{a_i \rightarrow \langle a_k, s_j \rangle}}{|\mathcal{A}_g|} \qquad (1)$$

[1]We assume that agents are willing to share their trust values with the algorithm.

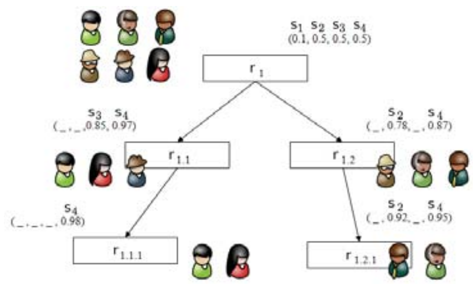


Figure 1: Example of trust spaces in role taxonomies

We denote the set of vector representations of agents – e.g., the trust space formed by agents – by:

$$TS = \{\widehat{a_k} = (\bar{s}_1, \bar{s}_2, ..., \bar{s}_n) \mid a_k \in \mathcal{A}_g\}$$

In a similar way, given a role $\langle r, \mathcal{E}_r \rangle \in R$, we can define a trust space for the agents that play that role:

$$TS_r = \{\widehat{a_k} = (\bar{s}_j, ..., \bar{s}_l) \mid s_i \in \mathcal{E}_r \text{ and } a_k \in ag(r)\}$$

The centroid of the trust space, $\widehat{TS_r} = (\widehat{s_j}, ..., \widehat{s_l})$ with $s_i \in \mathcal{E}_r$, represents how trusted the agents playing that role are regarding the interactions specified in $\mathcal{E}_r$. A trust space TS_r can be subdivided into groups of agents such that these groups fulfil the requirements of Definition 2 for new specializations of the role r. In figure 1 we give an example. In this figure, the values next to the roles represent the centroids of the vectors representing the agents that can enact those roles. A role may be specialized if a group of agents playing that role are more trustworthy regarding some of the interactions the role specifies. Each new specialized role defines its own trust space, which could then be subdivided again according to Def. 2 to define even more specialized roles.

3.2 Role Creation

Trust is a subjective measure. Agents will usually not perfectly agree on the expectations that others generate in the system. This implies that determining whether or not an existing role should be specialized and in which sense is not a straightforward task. We use the *K-means* clustering algorithm together with some post-processing of the output clusters for this task. Clustering algorithms in general, and K-Means in particular, divide data in different groups or clusters according to their similarity in a n-dimensional space. Accordingly, clusters in a trust space represent social agreement patterns on the behaviour of agents in the system.

Let O be an OMAS with a set of roles R and a role specialization taxonomy $\mathcal{RT} = (R, \rhd_r)$. In order to evolve the role taxonomy, the clustering algorithm is applied to each set TS_r with $\langle r, \mathcal{E}_r \rangle \in R$ being a node in the taxonomy $\mathcal{RT}$. On each execution, the algorithm returns a set of k clusters. A cluster c is a subset of the trust space TS_r and represents a pattern of behaviour for all the included agents. The cluster centroid $\widehat{c} = (\widehat{s_j}, ..., \widehat{s_l})$ with $s_i \in \mathcal{E}_r$ represents how trusted all the agents belonging to this cluster are regarding the interactions defined in role r.

The k clusters returned by the algorithm are candidates for the creation of new roles. For each cluster c we check the following properties in order to decide whether it should be included as a new role or not:

Property $p1$. Considering Definition 2, in order to include a new role in the taxonomy we need to guarantee that the role fulfils the ordering $\rhd_r$ with the role from which it has been created. Furthermore, we do not create roles with "bad" behaviours. We apply a threshold $\Theta \in [0..1]$ to assure that the expected utility of the interactions of the potentially new role is at least Θ. Given $\widehat{TS_r} = (x_1, ..., x_m)$ and $\widehat{c} = (y_1, ..., y_m)$, $p1$ is fulfilled if: $\exists i : y_i > x_i \wedge y_i \geq \Theta$.

Property $p2$. As we want to create roles persisting in time, we fix a threshold Υ to guarantee that a minimum number of agents are able to play the new role. The property is fulfilled if $|c| > \Upsilon$.

Algorithm 1 describes the process of role creation. Given a role r, the result of the algorithm is a set of new specialized roles R'.

Algorithm 1 *roleCreation* algorithm

Require: $\langle r, \mathcal{E}_r \rangle \in R$ {where $(R, \rhd_r)$ is a role specialization taxonomy}
$TS_r = \emptyset$
for $a_j \in ag(r)$ **do**
  $\widehat{a_j} \leftarrow calculateTrustVector(a_j)$ {with equation (1)}
  $TS_r \leftarrow TS_r \cup \{\widehat{a_j}\}$
end for
$C \leftarrow KMeans(TS_r)$ $\{k = |\mathcal{E}_r|\}$
$R' \leftarrow \emptyset$
for $c \in C$ **do**
  if $check(c, p1) \wedge check(c, p2)$ **then**
    $r_{new} \leftarrow clusterToRole(c)$
    $R' \leftarrow R' \cup \{r_{new}\}$
  end if
end for
return R'

3.3 Taxonomy Adaptation

Until this point we have put forward how a role taxonomy can be created from the trust values the agents provide. The created role taxonomy depends on the social dynamics of the system at a given interval of time. However, especially in open and highly volatile environments, this dynamics may change. Changes could be due to several reasons: agents could leave or enter the system or agents may change their behaviour over time (e.g., because of new internal goals).

To take such changes into account and to create a mechanism that remains useful over time, the role taxonomy should evolve and should adapt to the changes in the population of the system. In order to do that we implement an algorithm (Algorithm 2) that adapts the role taxonomy to the current system characteristics. This algorithm is repeated every t time steps. Starting with the root role (r_{root}), it applies the role creation algorithm to each role in the current taxonomy. The algorithm returns a new set of roles (R_{new}) for a new taxonomy ($R_{new}, \rhd_r$) that adapts to the current system population. It allows for: *i) Emergence*, new roles can be created as specialisations of existing ones; *ii) Deletion*, existing roles that are not identified as potential roles any more are deleted (together with all their child roles); and *iii) Maintenance*, a role is maintained but the set of agents assigned to that role may change (through *assignAgentsToRole*).

Figure 2 presents an example of a role taxonomy that has been created with the role evolution mechanism in two sequential executions. In this example, five agents participate in the system and the set of service type interactions is $\mathcal{S} = \{s_1, s_2, s_3, s_4\}$. Roles r_1 and r_2 have been created as specialisations of role r_{root} in a first execution of the algorithm. In a second execution these roles remain and a new role $r_{2.1}$ has been added as a specialization of role r_2. The algorithm assures that agents are assigned to roles regarding the expectations (trust) they have generated in the society as providers of the different service type interactions. For instance, the agent playing role $r_{2.1}$ has generated the highest expectations as a service provider for s_2 and s_3.

Algorithm 2 *roleAdaptation* algorithm

Require: $\mathcal{RT} = (R, \rhd_r)$ {the current taxonomy}
Require: $r \in R$
$R_{new} \leftarrow \emptyset$
$R' \leftarrow roleCreation(r)$
for $r_i \in R'$ **do**
  $assignAgentsToRole(r_i)$
  $R_{new} \leftarrow R_{new} \cup \{r_i\}$
  if $r_i \in R$ **then**
    $R_{new} \leftarrow R_{new} \cup roleAdaptation(\mathcal{RT}, r_i)$
  end if
end for
return R_{new}

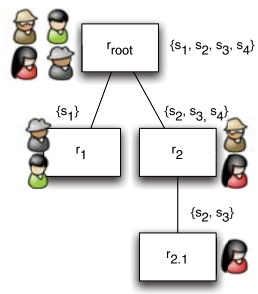


Figure 2: Example of a role specialisation taxonomy

4. BUILDING THE ARTIFACT

In order to build, evolve and make role taxonomies available to the agents in a system, some kind of service must be specified[2]. We use the concept of *coordination artifact* [15] as a way of incorporating the role evolution mechanism into an OMAS. A Coordination Artifact (CA) is conceived as a persistent entity specialised in providing a coordination service in a MAS.

The CA abstract model [14] is characterised by: (i) an usage interface, i.e. operations that allow agents to *use the artifact*; (ii) a set of operating instructions, that define how to use the artifact in order to exploit its coordination service; and (iii) a coordination behaviour specification, that describes the coordinating behaviour of the artifact. Artifacts are defined as computational entities that can be used by agents. They offer a specified functionality developed at designed time, and that should potentially serve agents' purposes somehow. The operations that a CA offers to the

[2]We assume that the environment is not a passive entity but can encapsulate services to support agent coordination or organisation.

agents should produce some desirable effects on the environment. In our case, the CA mission is to build and evolve a role specialisation taxonomy over time and make this information available to agents. For that purpose, the CA will offer the following usage interface operations:

- $getBestRolesForInteraction : \mathcal{S} \Rightarrow \mathcal{P}(R)$ provides the most specialised roles for a given service type interaction;
- $getAgentsForRoles : \mathcal{P}(R) \Rightarrow \mathcal{P}(\mathcal{A}g)$ provides the set of agents that play at least one of the roles in a given set of roles;
- $getRolesForAgent : \mathcal{A}g \Rightarrow \mathcal{P}(R)$ provides the set of roles a given agent plays in the system;
- $getTaxonomy$: provides the current role specialization taxonomy.

$\mathcal{P}(Z)$ denotes the power set of a set Z. These operations are used by the agents to receive information from the CA. Finally, the artifact requires another operation that allows to obtain the trust values from the agents participating in the system. This operation is defined as follows:

- $communicateTrust : \mathcal{A}g \times \mathcal{S} \times \mathbb{R}$.

An agent a can call $communicateTrust(b, s, v)$ indicating that its trust value regarding the capacities of agent b as a provider of the service s is v. It is certainly an interesting issue to study when and why agents should provide their trust values and, even more, the possibility to intentionally provide false trust values. However, in this paper we do not analyse such issues. We assume agents transmit their real trust values autonomously on a more or less regular basis.

The inner functioning of the artifact is fixed by the role specialization taxonomy adaptation algorithm we have introduced before. On a regular basis (each t time steps) the artifact recompiles the received trust values and executes this algorithm. Formally, given the role taxonomy $\mathcal{RT} = (R, \rhd_r)$ and the root role r_{root} in the OMAS at time t, the artifact substitutes the current taxonomy with a new taxonomy $(R_{new}, \rhd_r)$ (and assigns the agents in the OMAS to the roles in R_{new}) as follows:

$$R_{new} \leftarrow roleAdaptation(\mathcal{RT}, r_{root})$$
$$\mathcal{RT} \leftarrow (R_{new}, \rhd_r)$$

Whenever agents call any of the first four operations above the artifact returns the requested information corresponding to the current role taxonomy.

5. EXPERIMENTAL EVALUATION

The experimental evaluation of the presented work is based on the possible use agents make of the role taxonomy evolution artifact. In this section, we first introduce a trust model for agents that exploits role taxonomy information in order to infer expectation about future interactions. The use of the artifact may reduce the complexity of the agents' search of providers, driving them to better selections through the time. Afterwards, we describe the scenario we have used in our experiments and we show some test results to validate our work.

5.1 Using the artifact: A role-based trust model for agents

A trust model is normally used to endow agents with an internal representation of information about others, in order to choose appropriate partners to interact with in a MAS. Trust models aim at calculating expectations on the behaviour of others in particular interactions. Usually they rely on an agent's own experience in past interactions gathered through the time and/or using opinions from third parties – reputation mechanisms. We claim that there exists another source of information that may influence agents' reasoning about trust. In particular, roles in its descriptive facet can provide additional information since they may represent expectations on agents playing that roles in different interactions.

Algorithm 3 describes how an agent a uses the information provided by a role specialization taxonomy $\mathcal{RT} = (R, \rhd_r)$ together with its own experience about previously performed interactions in order to select an appropriate provider of a service $s \in \mathcal{S}$ it is interested in.

Algorithm 3 Algorithm describing an agent's selection process.

Require: s - the service agent a wants to use
Require: CA the coordination artifact with the current role specialization taxonomy $\mathcal{RT} = (R, \rhd_r)$
1: $R' \leftarrow getBestRolesForInteraction(s)$ $\{R' \subseteq R\}$
2: $\mathcal{A}_x \leftarrow getAgentsForRoles(R')$ $\{\mathcal{A}_x \subseteq \mathcal{A}_g\}$
3: $bestAgent \leftarrow localTrustEvaluation(\mathcal{A}_x, R', s)$
4: $perform(s, bestAgent)$

Following algorithm 3, firstly the agent uses the artifact to obtain the set of most specialized roles $R' \subset R$ for the interaction s. This set contains all roles $\langle r_1, \mathcal{E}_{r_1}\rangle \in R$ with $s \in \mathcal{E}_{r_1}$ and for which there exists no specialization regarding interaction s, that is, for which there exists no other role $\langle r_2, \mathcal{E}_{r_2}\rangle \in R$ with $\langle r_1, \mathcal{E}_{r_1}\rangle \rhd_r \langle r_2, \mathcal{E}_{r_2}\rangle$. In the second step, a uses the artifact to obtain the set of agents that play any role in the set R' in the system. In the third step, the agent employs its own individual trust model to select the most trustworthy agent – regarding its own experience – out of the agents playing at least one of the roles in R'. Finally, a performs the interaction with the agent it selected in step 3.

As it can be seen from this algorithm, using the role taxonomy may reduce the complexity of selecting service providers because the agents will only choose among a subset of all agents as possible candidates.

Regarding step 3, the local trust evaluation, in this paper we use a trust model that already combines information provided by role taxonomies with the agent's own experience. This model does not use reputation information. Nevertheless, other models using reputation could be employed. As suggested by Hermoso et al. in [7] this model is based on the assumption that *agents tend to behave similarly when enacting similar roles*. Using this assumption, an agent estimates the future behaviour of another agent in a certain situation by considering its past behaviour in "similar situations". That is, an agent can infer trustworthiness, even if it has (i) no direct past experience about another agents playing a specific role and, (ii) it cannot collect opinions from other agents either because the opinions from others are unreliable or none of the agents have yet enough proper experiences. Using this approach, agent a calculates a trust value $t_{a\to\langle b,r\rangle} \in [0..1]$ for each agent $b \in \mathcal{A}_x$ and each role $r \in R'$ that b plays in the system and selects the agent (b) with the highest value as the provider for the desired service. If an agent plays more than one role in R', then the average of the trust values for each role is chosen.

For the calculation of trust values $t_{a\to\langle b,r\rangle}$, we assume that agents do not only store their experiences with others regarding the provisioning of certain services, but also regarding the roles those agents play in the system. In particular, agents store their experience in form of confidence values $c_{a\to\langle b,r\rangle}$, denoting the recompiled confidence an agent a has in agent b playing role r. As a default value, e.g., a does not have any experience regarding agent b playing role r, we use 0.5. Based on the confidence values and the aforementioned similarity assumption, trust values are calculated using the following equation:

$$t_{a\to\langle b,r\rangle} = \frac{\sum\limits_{\langle b,r_j\rangle \in IS_a} c_{a\to\langle b,r\rangle} \cdot sim(r_j, r)}{\sum\limits_{\langle b,r_j\rangle \in IS_a} sim(r_j, r)} \tag{2}$$

where sim is a similarity function on roles and IS_a represents a's internal structure, that is, the set of confidence values it stores regarding agents playing certain roles in the system. Equation 2 calculates the trust agent a has in agent b playing role r as the weighted average of its confidence in b playing any role in the system, weighted by the similarity of that roles to the role r.

We estimate the similarity between two roles on the basis of the role taxonomy provided by the artifact through its operation $getTaxonomy$ as follows:

$$sim(x,y) = \begin{cases} 0 & , \, if \; not \; (x \rhd_r y \; or \; y \rhd_r x) \\ 1 - \frac{h}{h_{MAX}} & , \, otherwise \end{cases} \tag{3}$$

where x, y are roles in $\mathcal{RT}$, h is the number of hops between x and y in the same branch of the taxonomy, and h_{MAX} is the longest possible path between any pair of elements in the same branch of the tree. Although other more sophisticated similarity functions could be used [5], for the sake of simplicity we used this rather simple function to illustrate our approach.

5.2 News Providing Scenario

We have chosen a news providing scenario to test our coordination artifact. It deals with a classical problem of client/provider flow. Users periodically want to receive news about topics they are interested in. Topics are organised in categories – they could be seen as concept clouds. News providers may provide news of different categories or may be specialised on a subset of those categories. The problem for the users consists on selecting an adequate news provider whenever he is interested in news about one or more topics (see Fig.3). We assume that users do not know *a priori* the types of news that each provider contains. We will use our coordination artifact in order to evolve different news provider roles that will help users agents when selecting appropriate providers. We have taken 20 categories defined by Google News [3] for our experiments.

[3] http://www.google.com/support/webmasters/bin/answer.py?hl=en&answer=42993.

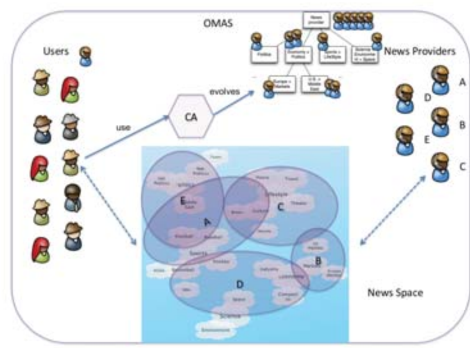


Figure 3: Schema of the News Providing scenario.

The particular problem of news provider selection has been widely studied in Information Retrieval (IR). However, most IR approaches tackle the problem by using lexical-oriented techniques that classify providers to categories based on the lexical content of the documents they publish. We claim that our approach obtains a similar result but is also intrinsically different. Our coordination artifact does not classify news providers as to the content of the documents but with respect of the perception the current user population has regarding the usefulness of certain providers for certain categories of news. This user-centred approach (instead of information-centred approach) does not only take into consideration the lexical match of news to the topics of interest of users. It will also take into account other properties of news providers that the current population of users appreciate (e.g., the reliability of the news, the price of the service, etc.). In this sense, our approach adapts better to the needs and desires of a current population of users than a predefined classification based on lexicographic techniques.

5.2.1 *Experimental setup*

To determine the effectiveness of our artifact we have carried out a set of experiments where we simulated the news provider scenario. In each experiment we use 100 user agents and 100 news providers. The set of service type interactions $\mathcal{S} = 20$ coincides with the 20 news categories. We simulate the expertise of a provider agent a for the service type interaction s as a normal distribution $N(\mu_{a,s}, \sigma_{a,s})$, where the mean $\mu_{a,s} \in [0..1]$ and the standard deviation $\sigma_{a,s} \in [0..0.05]$ are randomly chosen in the corresponding intervals. Experiments were run for 50 time steps. In each time step, each user agent has to select one provider agent for each $s_i \in \mathcal{S}$. That is, each time step includes 2000 interactions (100 user agents times 20 categories). After each time step, user agents evaluate the utility of the interactions and update their local trust model. We use the normal distributions describing the expertise of providers to simulate the utility a user perceives when requesting news about a certain category from a provider. Every time a user requests news for a given category, a result utility value is randomly generated from the corresponding distribution by using the *polar method* [12]. We analyse how the utility evolves for systems with different characteristics. We compare the evolution of the systems when the coordination artifact is used to the evolution when the artifact is not available. In all experimental runs with the coordination artifact, the initial taxonomy just contains the root role ($\mathcal{RT}^{t_0} = (\{r_{root}\}, \triangleright_r)$) and all agents are assigned to this role. The role taxonomy evolution algorithm is executed after each single time step. In the runs without the artifact, agents' selection of appropriate providers is only based on their past experiences. That is, if agent a requires a provider for service s, it selects the agent that has provided it the highest utility for that service in the past. In order to find new and possibly better providers agents also implement a simple exploration policy, where providers are chosen randomly in about 5% of the cases. Each experimental run has been repeated 10 times with different random seeds, being the presented results an average of them.

5.2.2 Results

In a first set of experiments we analyse how the use of the coordination artifact may help agents to improve their utility. It shows the results for different values for the parameters Θ and Υ. The figures show how increasing these thresholds, forming roles becomes more complicated and the artifact can provide less help to agents. In fact, in figure 4(c) new roles are hardly created and, therefore, looking for partners becomes more difficult.

In Fig. 4 we have assumed that user agents have the same utility function. To analyse the effect of the artifact if the agents have different perceptions regarding the trustworthiness of news providers, we conducted another experiment. In this case we define two different types of users, namely TYPE A and TYPE B users. TYPE A users share their utility function, e.g., the utility value they receive for a new providing interaction is drawn from the normal distribution from the news provider. TYPE B users are "strangers" and introduce some noise in the system. To simulate this, the utility these users receive after using the service of a news provider is randomly chosen from the interval [0..1]. We have conducted two experiments: one with 75% of TYPE A users and (25% TYPE B) and another with 25% of TYPE A users (75% TYPE B). The results are presented in figure 5(a). The results indicate that the higher the agreement among the agents the higher is the improvement when the coordination artifact is used. The reason is that a higher agreement among the agents regarding the trustworthiness of news providers create more crisp clusters in the clustering algorithm and allows for the creation of richer role taxonomies. A richer role taxonomy implies more information for agents and leads to an improvement in the provider selection process. In this context it should be noted that if there is no agreement at all among the agents regarding the behaviour of others, no roles will be created. In this case the performance of the system with the coordination artifact will degrade to the performance of the system without the artifact.

In the last set of experiments we analysed the effect of the artifact in a more dynamic environment, in particular, if user agents enter and leave the system at any time. Figures 5(b) and 5(c) show the results. In 5(b) we have randomly chosen a 10% of user agents that leave at each time step, being substituted by the same number of "newcomer" agents. In 5(c) this percentage is increased to a 30%. Both experiments show that even when significant changes in the population occur, the use of the artifact provides advantages. This is because the roles created to specify provider behaviours persist over time and help agents to make their decisions. This holds especially for newcomers that cannot rely on their own experiences when selecting providers of service type interactions.

6. CONCLUSIONS

In this work we put forward a coordination artifact for OMAS to help agents to take more informed decisions regarding partner selection. This artifact evolves a role taxonomy, and assigns agents to roles based on the trust agents have on other agents as partners in different types of interactions. This information can be used by agents to better estimate the behaviours of others as potential partners in future interactions. We define roles as descriptive entities, providing *expectations* about the behaviour of agents in certain types of interactions, rather than their normative facets. We present some experiments that show that the use of the proposed artifact helps agents to select better partners for their interactions than selection processes based only on agents' own experience.

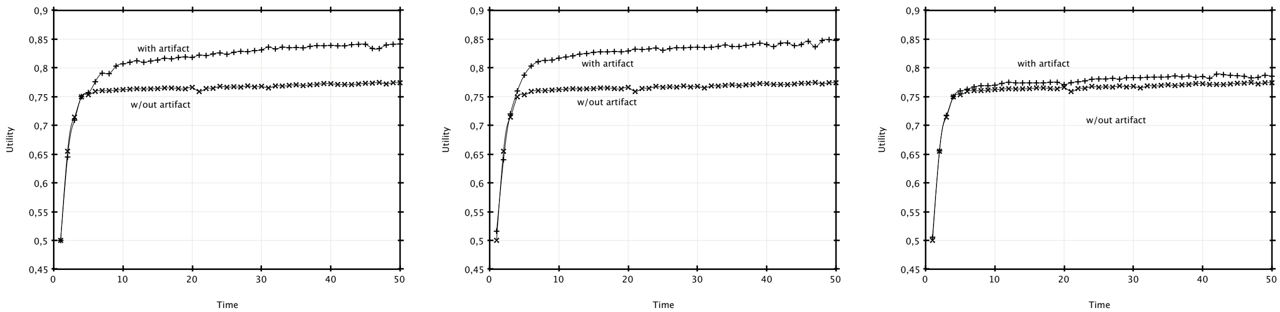
Most work in the field of OMAS has been done intending to regulate OMAS with prescriptive structures. In [1], the author puts forward an infrastructure for dynamic protocol specifications, that is, specifications that may be changed at runtime by agents participating in an OMAS. This approach is similar to ours in the sense that it tries to "organise" the system at runtime from the agents that populate the system. However, we do not impose constraints on agents on decision-making processes. In [4], Esteva et al. argue that OMAS can be designed and implemented as electronic institutions. However, that approach does not support dynamic changes in the system, such as changes in the population. Those changes can be hardly tackled at design time.

Some interesting works have been published about structural adaptation. In [13], the author presents a decentralised approach for structural adaptation in MAS. The method enables the agents to implicitly adapt their structural relationships to improve task allocation processes. Other works allow a MAS to change its organisation during execution. For instance, in [2] the authors present a framework to allow MAS organisations to re-organise at runtime, while in [8] the author presents a mechanism to dynamically adapt an organisational model to environmental fluctuation in max flow networks. There is also much literature about re-organization of organisation structures in MAS [6, 3], but none of them have exploited the trust relationships that may emerge in a system as a support for re-organisation.

Regarding coordination artifacts for trust management, in [10] the authors present reputation as a collective process, using an artifact in order to publish and provide some objective evaluations that agents calculate. Our approach goes beyond this, since our role evolution mechanism allows for the aggregation of subjective trust evaluations of the agents so as to dynamically evolve the role taxonomy. Finally, much work has been done in the field of trust and reputation mechanisms to endow agents with more information when making decisions (e.g. [16, 11]). Regarding such models, the proposed role taxonomies can be considered as an additional dimension that may help to estimate trust.

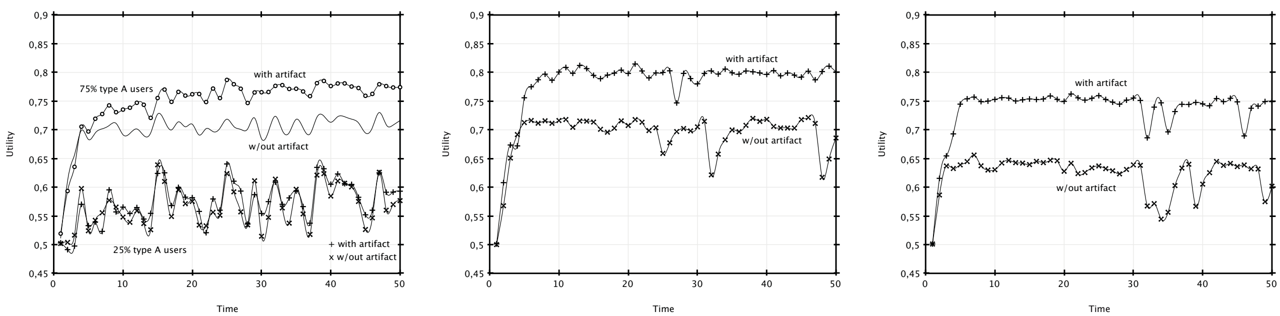
As future work we plan to enrich our experiments by allowing dynamic changes on providers. We are interested in studying how providers joining and leaving the system affect the overall utility of the system.

7. REFERENCES



(a) $\Theta = 0.7$ and $\Upsilon = 3$ (b) $\Theta = 0.85$ and $\Upsilon = 3$ (c) $\Theta = 0.85$ and $\Upsilon = 7$

Figure 4: Average results for different values of Θ and Υ



(a) Heterogeneity TYPE A/B USERS (b) 10% changes on users' population (c) 30% changes on users' population

Figure 5: Average results for OMAS with different characteristics

[1] A. Artikis. Dynamic protocols for open agent systems. In *AAMAS '09*, pages 97–104, Richland, SC, 2009. IFAAMAS.

[2] S. A. Deloach, W. H. Oyenan, and E. T. Matson. A capabilities-based model for adaptive organizations. *Autonomous Agents and Multi-Agent Systems*, 16(1):13–56, 2008.

[3] V. Dignum, F. Dignum, and L. Sonenberg. Towards dynamic reorganization of agent societies. In *In Proceedings of Workshop on Coordination in Emergent Agent Societies*, pages 22–27, 2004.

[4] M. Esteva, J. A. Rodríguez-Aguilar, J. L. Arcos, C. Sierra, and P. García. Institutionalising open multi-agent systems. Technical report, IIIA-CSIC. IIIA Research Report 2000-01, 2000.

[5] P. Ganesan, H. Garcia-Molina, and J. Widom. Exploiting hierarchical domain structure to compute similarity. *ACM Transactions on Information Systems*, 21(1):64–93, 2003.

[6] Z. guang Wang, X. Liang, and Q. Zhao. Adaptive mechanisms of organizational structures in multi-agent systems. In *PRIMA - LNAI*, volume 4088, pages 471–477. Springer-Verlag, 2006.

[7] R. Hermoso, H. Billhardt, R. Centeno, and S. Ossowski. Effective use of organisational abstractions for confidence models. In *ESAW*, volume 4457 of *LNAI*, pages 368–383, Berlin, 2007. Springer.

[8] M. Hoogendoorn. Adaptation of organizational models for multi-agent systems based on max flow networks. In *IJCAI*, pages 1321–1326, 2007.

[9] B. Horling and V. Lesser. A Survey of Multi-Agent Organizational Paradigms. *The Knowledge Engineering Review*, 19(4):281–316, 2005.

[10] J. F. Hübner, L. Vercouter, and O. Boissier. Instrumenting multi-agent organisations with artifacts to support reputation processes. In *COIN 2008*, pages 96–110, Berlin, Heidelberg, 2009. Springer-Verlag.

[11] T. D. Huynh, N. R. Jennings, and N. R. Shadbolt. FIRE: An integrated trust and reputation model for open multi-agent systems. In *ECAI'04*, 2004.

[12] D. E. Knuth. *The Art of Computer Programming - Seminumerical Algorithms vol.2*. Addison-Wesly, 1998.

[13] R. Kota, N. Gibbins, and N. R. Jennings. Self-organising agent organisations. In *AAMAS'09*, pages 797–804, Richland, SC, 2009. IFAAMAS.

[14] A. Omicini, A. Ricci, and M. Viroli. Artifacts in the a&a meta-model for multi-agent systems. *Autonomous Agents and Multi-Agent Systems*, 17(3):432–456, 2008.

[15] A. Omicini, A. Ricci, M. Viroli, C. Castelfranchi, and L. Tummolini. Coordination artifacts: Environment-based coordination for intelligent agents. In *AAMAS'04*, volume 1, pages 286–293. New York, USA, ACM, 2004.

[16] S. D. Ramchurn, C. Sierra, L. Godó, and N. R. Jennings. A computational trust model for multi-agent interactions based on confidence and reputation. In *Proceedings of 6th International Workshop of Deception, Fraud and Trust in Agent Societies*, pages 69–75, 2003.