Organising Multiagent Systems by Using Role Clustering Mechanisms *

Ramón Hermoso, Holger Billhardt, Roberto Centeno, and Sascha Ossowski

Centre for Intelligent Information Technologies (CETINIA)
University Rey Juan Carlos, Madrid (Spain)
{ramon.hermoso, holger.billhardt, roberto.centeno, sascha.ossowski}@urjc.es

Abstract. In recent years the study of organisational aspects of multiagent systems has gained much importance. In previous work we have shown that enhancing trust mechanisms with organisational information, and in particular with role taxonomies, usually allows agents to take better decision regarding which acquaintances to interact with. This paper addresses the parallel problem, exploring how organisational structures may evolve over time using the information from the agents' trust models. We present a mechanism based on clustering techniques capable of detecting behavioral patterns in MAS, thereby identifying new roles that dynamically extend the role taxonomy. We present experimental results showing that this extension leads to an improvement of the agents' decision making processes when compared to static organisational structures.

1 Introduction

Endowing MultiAgent Systems (MAS) with an organisational flavour has been extensively studied in recent years. It is commonly accepted that MAS designed with organisational structures allow tackling complex problems from an easier perspective [13]. Most of the times organisational structures are conceived as static design patterns that regulate agent behaviour. Nevertheless, lately the importance of openness and dynamism in MAS is growing, calling for structures that dynamically adapt over time to changing circumstances [19, 1]. In addition, most of the work about organisations in MAS is centered on macro-level issues [8, 2, 11, 3] while the questions of how autonomous agents in open systems deal with them appears to be secondary.

Even if we assume that agents do not transgress organisational prescriptions, it is important to notice that by definition autonomous agents should always be given a certain freedom of choice. Setting out from this premise, we focused our previous work [6, 5] on trust mechanisms that agents use to produce expectations about the behaviour of others in order to better choose partners to interact with

* The present work has been partially funded by the Spanish Ministry of Education under project TIN2006-14630-C03-02 (FPI grants program) and by the Spanish project "Agreement Technologies" (CONSOLIDER CSD2007-0022, INGENIO 2010)

during task performance. Many works have been presented in this field [9, 16, 17], but these mechanisms are not related with other studies, that is, they are not used as a means to develop other more complex mechanisms. We claim that the information stored in agents' internal structures has value not only in agents' decision processes, but also to help all participants in the system.

This work puts forward a mechanism that makes use of the information managed by the agents' trust models so as to create and evolve organisational structures, and in particular role hierarchies. This will finally help agents to act in a even more efficient way in the system. For this purpose, we propose a multidimensional clustering algorithm to capture behavioural patterns among agents.

The paper is organised as follows: in section 2 we specify a task-oriented MAS as a type of MAS we will use to describe our mechanism. Furthermore, we will present the organisational abstractions with which we endow the MAS to develop our mechanism. In section 3 we present our mechanism based on clustering; section 4 describes the experiments we have carried out; in section 5 we discuss and compare our proposal with related work; and finally section 6 summarizes the presented work and gives some guidelines of future work.

2 Organisational Mechanism

This section puts forward the type of domains and systems that our mechanism targets, as well as its basic "ingredients". We first specify task-oriented MAS, where agents are assigned tasks which they can either perform on their own or delegate to others. Then, we describe the organisational structures that have to be present in the system to run our mechanism. Finally we present a trust model that makes use of these organisational structures to help agents when selecting partners to interact with.

2.1 Task-oriented MAS

The environment we use to design the mechanism presented in this work is based on Task-oriented Multiagent Systems (T-MAS) which can be specified as follows:

Definition 1. *A T-MAS is a tuple* $TM = \langle \mathcal{A}_g, \mathcal{T}, \mathcal{U}, init \rangle$*, where:*

- $\mathcal{A}_g$ *is a set of agents participating in the MAS;*
- $\mathcal{T}$ *is a set of tasks that can be performed by agents;*
- $\mathcal{U} : \mathcal{T} \times \mathcal{A}_g \rightarrow \mathbb{R}$ *is an utility function that establishes a real value for an agent performing a certain task;*
- $init : \mathcal{T} \rightarrow \mathcal{A}_g$ *is a total function that initially assigns tasks to agents in the MAS.*

Notice that for the purpose of this paper we do not consider resource-limitations of agents; the utility of a task t is determined by the agent that performs it (i.e.

how “skillful” it is in performing that specific task) and does not take into account the amount and/or the type of the other tasks that the very same agent may have agreed to execute simultaneously. However, the reader should take into account that, although this excludes some interesting settings, it still covers a number of relevant domains (see Section 4 for an example). For the present work, we also consider the set of agents and tasks to be fixed.

Definition 2. *The* result *of a run of a T-MAS TM is a new task distribution represented by a total function* $res : \mathcal{T} \rightarrow \mathcal{A}_g$.

2.2 Organisational Information

We are interested in how organisational information may help agents to agree faster on “good” task distributions in a T-MAS. Many organisational concepts can be found in the MAS literature: *roles*, *interactions*, or *norms* are just some of them. Still, the contents and scope of each of these terms is sometimes quite blurry. The mechanism presented in this paper is based on the use of the concept of *roles*, which we conceive from the point of view of an observer, i.e. as a set of *expectations* regarding the behaviour of agents that play a certain role.

Definition 3. *Let* $\mathcal{R}$ *be a set of role identifiers. A role is a pair* $\langle r, \mathcal{E} \rangle$ *where*

- $r \in \mathcal{R}$ *is the role name;*
- $\mathcal{E} : \mathcal{T} \rightarrow \mathbb{R}$ *is a partial function representing a set of expectations regarding the quality with which tasks be performed. This function is often written as a vector.*

Notice that in the definition of T-MAS any agent can, in principle, perform any task (of course, the utility with which it executes it may be very low or even zero).

Definition 4. *A role taxonomy is a structure* $\mathcal{RT} = (R, \preccurlyeq)$ *consisting of a set* R *of role identifiers and a partial order* $\preccurlyeq$ *on* R.

A role taxonomy structures its elements by establishing a relation $\preccurlyeq$ of specialization; that is, given two different roles $r_1, r_2 \in R$ and $r_1 \preccurlyeq r_2$, then r_1 is a specialization of r_2.

It is straightforward to define a measure of similarity on the basis of a role taxonomy. For the time being it is not relevant how role similarity is exactly defined. We just assume that some such function exists.

Definition 5. $sim : \mathcal{R} \times \mathcal{R} \rightarrow \mathbb{R}$ *is a function that given two roles returns a value representing the similarity between them.*

2.3 Trust mechanism based on confidence

In this section we summarize a trust model that takes advantage of the organisational structures of a T-MAS, already presented in previous work [6, 5]. The trust model is based on the assumption that *agents tend to behave similarly when enacting similar roles in similar tasks.* Using this assumption, an agent is able to assess (to a certain extent) the future behavior of another agent in a certain situation by considering its past behavior in "similar situations". That is, an agent can infer trustworthiness, even if it has, first, no direct past experience about a situation and, second, it cannot collect opinions from other agents either because the opinions from others are unreliable ("liars") or none of the agents has enough proper experience. So, even though we adhere to the standard notion of *reputation* and *confidence* used in literature [9, 14], in this paper we will exploit organisational information with regards to the latter.

In our model, a confidence value $c_{A\rightarrow\langle B,R,T\rangle}$ is built up from A's past tasks requested to agent B playing – the latter – role R and performing tasks of type T. We call LIT – *Local Information Table* – the agent's data structure dedicated to store confidence values for past tasks with any counterpart the agent has requested. Each entry corresponds to a *situation*: an *agent* playing a specific *role* in a particular *task*. LIT_A denotes agent A's LIT. Each entry in a LIT consists of:

- the Agent/Role/Task identifier $\langle X, Y, Z\rangle$,
- the confidence value for the situation ($c_{A\rightarrow\langle X,Y,Z\rangle} \in [0..1]$),
- a reliability value ($r_{A\rightarrow\langle X,Y,Z\rangle}$) that measures how certain an agent is about its own confidence in situation $\langle X, Y, Z\rangle$.

If an agent participates in a task T with agent B playing role R, the corresponding entry in its LIT will be updated as follows:

- **confidence value**: let $g_{\langle X,Y,Z\rangle} \in [0..1]$ denote the evaluation value an agent A calculates for a particular experience with the agent X playing role Y in the task of type Z. In our work, we use the following equation to update confidence:

$$c_{A\rightarrow\langle X,Y,Z\rangle} = \epsilon \cdot c'_{A\rightarrow\langle X,Y,Z\rangle} + (1-\epsilon)\cdot g_{\langle X,Y,Z\rangle}, \tag{1}$$

  where $c'_{A\rightarrow\langle X,Y,Z\rangle}$ is the confidence value in A's LIT before the task is performed and $\epsilon \in [0..1]$ is a parameter specifying the importance given to A's past confidence value.
- **reliability value**: we calculate reliability by using the approach proposed by Huynh, Jennings and Shadbolt [9]. This approach takes into account the number of interactions a confidence value is based on, and the variability of the individual values across past experiences.

The counterpart (partner to perform the task) selection of an agent A in a task T that requires a role R is guided by the trustworthiness value $t_{A\rightarrow\langle B,R,T\rangle} \in$

[0..1] for each known agent B. It is calculated from A's LIT as follows:

$$t_{A\rightarrow\langle B,R,T\rangle} = \frac{\sum\limits_{\langle X,Y,Z\rangle\in LIT_A} c_{A\rightarrow\langle X,Y,Z\rangle} \cdot w_{A\rightarrow\langle X,Y,Z\rangle}}{\sum\limits_{\langle X,Y,Z\rangle\in LIT_A} w_{A\rightarrow\langle X,Y,Z\rangle}} \tag{2}$$

$w_{A\rightarrow\langle X,Y,Z\rangle}$ is the weight given to agent A's confidence on situation $\langle X,Y,Z\rangle$. The weights combine the confidence reliability with the similarity of the situation $\langle X,Y,Z\rangle$ to the target situation $\langle B,R,T\rangle$ in the following way:

$$w_{A\rightarrow\langle X,Y,Z\rangle} = r_{A\rightarrow\langle X,Y,Z\rangle} \cdot sim(\langle X,Y,Z\rangle, \langle B,R,T\rangle) \tag{3}$$

The similarity function $sim(\langle X,Y,Z\rangle, \langle B,R,T\rangle) \equiv sim(\langle I_1\rangle, \langle I_2\rangle)$ (to simplify notation, I_1 is *situation 1* and I_2 is *situation 2*) is computed as the weighted sum of the similarities of the individual elements (agent, role and task) as it is shown in the following equation:

$$sim(\langle I_1\rangle, \langle I_2\rangle) = \begin{cases} \beta \cdot sim_R(R,Y) + \gamma \cdot sim_I(T,Z), \; if \;\; B = X \\ 0, \; otherwise \end{cases} \tag{4}$$

where $sim_R(R,Y)$, $sim_I(T,Z) \in [0..1]$ measure the similarity between roles and tasks, respectively, and β and γ, with $\beta + \gamma = 1$, are parameters specifying the sensibility regarding the individual similarities. $sim_R(R,R')$ and $sim_I(I,I')$ can be determined by some measure of the *distance* of concepts in the corresponding taxonomies. We use the simple measure

$$sim_R(x,y) = sim_T(x,y) = 1 - \frac{h}{h_{MAX}} \tag{5}$$

where x, y are either roles or tasks, h is the number of hops between x and y in the corresponding taxonomy, and h_{MAX} is the longest possible path between any pair of elements in the hierarchy tree. Other functions have been described in [4].

3 Evolving role taxonomies

After giving some introductory aspects of our work, now we attempt to explain that given a T-MAS: *i)* if we endow it with organisational concepts such as roles, and the taxonomy that structures them, and *ii)* allow agents to use a trust model that deals with this organisational structures to assess trust, then the system will obtain some advantages:

1. On the other hand agents use the information provided by the trust model to reason about which agents are better to delegate tasks among those which can play well a specific role;
2. The T-MAS may evolve their organisational structure - roles and role taxonomy - according to the trust network created in the system.

Both issues deserve an special interest, since although we have mentioned before any agent may play any role, roles will be created and will evolve capturing a behavioral pattern of agents playing them. For example, the role *Surgeon* will be created to fill the gap existing among the *Physicians* that are good - and accordingly trusted by others - at operating and those which do not reach a minimum level of quality to operate. Nevertheless, among surgeons, some of them could be better than others for some kind of specific operations, or even for the same operation. It is for this reason that a trust model help the agents to select to whom delegate a task, but also, and not less important, allows discovering if new roles should be created. In this section we cover this latter issue.

During the execution of a T-MAS, there are two different elements that can evolve:

- *Agents*. Agents belonging to a T-MAS may change their behavior at their will; and
- *Organisational structures*. Role taxonomies may evolve over time due to agents' behavioral changes.

We propose in this work to use clustering methods to capture behavioral patterns of agents performing tasks, thus making possible the creation of new roles - if needed - as specification of existing ones. The trust values that agent A holds for others in its LIT are based on confidence values ($c_{A \rightarrow \langle X,Y,Z \rangle}$) calculated from past delegated tasks. These trust tuples can be conceived as a n-dimensional trust space as follows:

$$\begin{aligned} TS_{R_i} = \{(c_1, c_2, \dots, c_n); c_k \in \mathcal{C}_{A_r \rightarrow \langle R_i, I_k \rangle} \\ 0 \leq c_k \leq 1 \text{ , } I_k \in \mathcal{T}, A_r \in \mathcal{A}\} \end{aligned}$$

where $\mathcal{C}_{A_r \rightarrow \langle R_i \rangle}$ is a set containing confidence values stored by agent A_r, belonging to the set of agents $\mathcal{A}_g$ that are members of the T-MAS, related to any type of task - from T_1 to T_n - with counterparts playing role R_i. Thus, organisational information is subdivided in groups of *role specializations* for groups of tasks, providing some extra information when agents have to select partners to interact with.

In figure 1 is described how new roles are extended according to the agents enacting existing roles. Values next to the roles represent expected confidence values for every task in the system - in this example t_1, t_2, t_3 and t_4 - that agents should have if they want to play them. Thus, from the primitive role R_1 two new roles are created - $R_{1.1}$ and $R_{1.2}$. These new roles are more specialized than R_1, since they contains high expected capabilities (see definition 3) for fewer each time group of tasks. Therefore we can observe that role $R_{1.1}$ could be suitable to perform task t_2, t_3 and t_4 while $R_{1.2}$ seems to be better for tasks t_1 and t_4. In a real-life example it has sense to specialize from role *doctor* those agents which operate with high quality nose and cheekbone fractures from those which operate with high quality cardiovascular lesions. In the same way we can observe how there are some agents that are better than the rest in $R_{1.1}$ performing task t_4, fact that causes the creation of a new role.

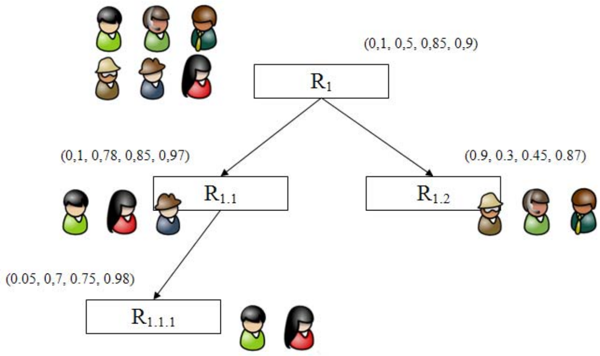

Fig. 1. Example of role taxonomy evolution

3.1 Trust-based Multidimensional K-Means

To specialize roles we use a multidimensional *K-means* algorithm [12, 10], where k represents the number of clusters to be made in each execution. The algorithm will take as input a set of points $(x, y) \in TS_{r_i}$ for each role $r_i \in \mathcal{RT}$ and each agent $a_j \in \mathcal{A}_g$ that may enact r_i in any task $t_k \in \mathcal{T}$ with available information, and will return a group of clusters. A cluster mean point represents mean behaviour for all the agents belonging to it (how they are trusted playing r_i in any kind of task in the system), and the whole cluster represents a pattern of behavior for all the agents included. Note that a point in the n-dimensional space represents the behavioral pattern of an agent for every type of task in the system. For example, in Figure 1, the mean behavioral pattern to play any role corresponds to the centroid of the cluster; that is, the mean behavior of the agent representing the centroid point in the four different tasks.

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, the creation of a new role in the taxonomy means that some of the agents share a close similarity for a smaller subset of tasks; i.e. all physicians know basic notions about how to operate, but just a few of them do it properly. Note that this similarity with which the mechanism creates new roles will be used by agents to better estimate expectations on others when utilizing their trust model presented in section 4.1.

Nevertheless, there are cases in which could be more useful not to create a new role:

- When the average of trust that agents contained in the cluster is very low for all the possible tasks. In our experiments consider a confidence less than 0.5 to be low to create a new cluster.
- When the number of agents in a cluster is under certain threshold could be suitable not to create the role; i.e. when only one agent forms the cluster. In our experiments this threshold is 3 agents.

4 Experiments

In this section we explain the domain we have used to test our mechanism. It is based on Information Retrieval (IR) systems and consists of a group of agents that make queries to others about some kind of information. Requested agents have to reply with a set of documents. Utility is measured as the number of documents found over the total number of relevant documents existing in the system for a specific query. We have followed the model proposed by Zhang et al. in [18] to implement the system.

4.1 T-MAS for IR domain

From the definition given in section 2.1, a possible instance for our scenario is the following:

1. Let $\mathcal{A}_g$ be a set of agents in the IR network;
2. Let $\mathcal{Q} \equiv \mathcal{T}$ be a set of queries that agents receive;
3. Let $\mathcal{U}_x : \mathcal{A}_g \times \mathcal{Q} \to \mathbb{R}$ represents the utility function for a requested query. In our case, this function is determined by the number of relevant documents found over the total number of relevant documents in the system: $\mathcal{U}_\S(a_x, q_i) = \frac{RD_x}{RD_{all}}$, where RD_x is the number of relevant documents found in $a_x \in \mathcal{A}_g$ and RD_{all} is the total number of relevant documents existing in the system for q_i; the mapping between queries and relevant documents is available using $\mathcal{J}$;
4. Let $init : \mathcal{Q} \to \mathcal{A}_g$ be a task assignment function that assigns queries to agents.
5. Let $res : \mathcal{Q} \to \mathcal{A}_g$ be a function that represents a final query performance assignment.
   Other elements must also be introduced:
6. Let $\mathcal{D}$ be a set of available documents in the IR network; $|D|$ will denote the number of documents in the IR network.
7. Let $\mathcal{J} : Q \to 2^{|D|}$ represents a judgements function, that assigns a set of relevant documents to each possible query in $\mathcal{Q}$;
8. $\mathcal{U} : (\mathcal{A}_g)^* \to \mathbb{R}$ is the global utility function, that we use to measure how good is the system. We will use an aggregation function to measure the overall utility of the system: $\frac{\sum \mathcal{U}_x}{|\mathcal{A}_g|}$. This function will have as input the function res since the latter represents who has perform each query in the system[1].

In order to perform the evaluation of our mechanism we have developed a tool - an extension of a previous tool [5] - called IR-TOAST that allows the user to generate a IR-oriented T-MAS and observe the results of applying our evolutive mechanism over time with different configurations.

[1] Note that in this case, the global utility function is aligned with the agents' utility function, thus an increase in the agents' utilities entails an increase in the system global utility

In section we explained that our trust model based its efficiency on agents' local information, through which they can infer expectations about others' behavior. In order to do this inference, agents may calculate similarities between role and between tasks using taxonomies. However, in the instance of the specification of T-MAS for IR domains we do not include a task taxonomy - in this case query taxonomy. Creating a new role in the role taxonomy entails not only that agents behave similarly enacting that role but also that if the role is suitable for several different queries if agents are "good" (high expectations on them) playing the new role, they are also similar when performing the same query with ancestor roles[2]. That is, if an agent has no queries stored in its LIT about a new role, it can infer the behavior of agents that are ready to play it from the information about ancestor roles similar to the new one.

Some considerations that should be taken into account are:

- Agents send queries to only one provider;
- A provider cannot attend simultaneously more than one query;
- Although a T-MAS consists on a group of agents performing tasks, in this instance we have created a task-delegation scenario, where agents are assigned tasks - queries - and they need to find any other agent to send the query - delegation - and so obtain the relevant documents. This domain could be considered as a service-oriented domain as well, where agents to which the system assigns queries would take the role of *customer* and those who are requested would take the role of *provider*.
- A top role is needed to run the system. We use the role *Knowledge Actor* to initialize the role taxonomy. Next roles are created following the clustering algorithm.

4.2 System bootstrapping

In the following items we describe the bootstrapping flow when the system starts running:

1. Documents have to be assigned among agents. For that we establish a parameter ω that establishes the percentage of documents in a category that an agent must have to be considered as an expert. Thus, we make sure that every agent will be, at least, expert in a category and also that utility values of best providers are high. We also introduce some noise in the agents' collections.
2. Once agents have their own document collection the system generates the possible types of queries that can be assigned. Queries will be created at the same time as judgments list. Note that queries have a random name and have no semantics.
3. Once the system knows the different queries that can be evaluated, then it assign them to the agents participating in the system. This process is randomly made with an uniform distribution.

[2] By ancestor roles we denote the relation $r_1 \preccurlyeq r_2$ where r_2 is an ancestor of r_1

4.3 Simulation setup

The set $\mathcal{D}$ of documents has been collected from Yahoo web page [7]. We collected documents from different overlapped categories, such as sports, cycling, movies, television, news, etc.

It is also necessary to give an initial setup for the experiments. The graphics that can be observed in figure 2 are the average of 5 different runs with the same setup. During each experiment the system randomly assigns 15000 queries - only 20 of them are different; the rest are repeated queries - uniformly among 15 different agents and 500 documents per agent. The algorithm of clustering has been used in a cyclical way every 1000 queries over the total. We are currently working on more complex politics for deciding when to re-organise the role taxonomy, such as context-aware mechanisms.

We have compared how evolve the system using four scenarios (based on different trust models and clustering mechanism). Note that trust models are a seminal piece for agents to reason about what other agents they should trust to delegate - or request - their queries. The following are the different scenarios used:

– *Basic model* - (*BM*). In this scenario agents use a trust model that focuses on the local information that agents have in their LITs to select the agent to whom send the query. They select among the agents that have behave better for the same query in the past. No clustering is made;
– *Inference Model* - (*IM*). This scenario allow agents to use the trust model described in section 4.1. No clustering is made;
– *Inference Model Evolving* - (*IMe*). In this scenario agents use the inference model but the system also runs the clustering mechanism to learn new roles;
– *Inference Model Evolved* - (*IMev*). In this scenario agents use the inference model using the previously learned role taxonomy.

4.4 Results

The results of the experiments performed show how our clustering mechanism improve the overall utility of the system (see figure 2). As it was expected, *IMev* entails profitable advantages, above all, at bootstrapping, since lack of information in agents' LITs make the use of the role taxonomy (already evolved) very useful. It is reasonable to think that *IMe* is also a good model as well, since the curve grows faster than in *BM*.

5 Related Work

Much work has been done in the field of trust and reputation mechanisms to endow agents with more information when taking decisions [14, 15, 17, 16, 9]. The difference with our approach is that those mechanisms have not been embedded in other mechanisms as we have done with the clustering mechanism for

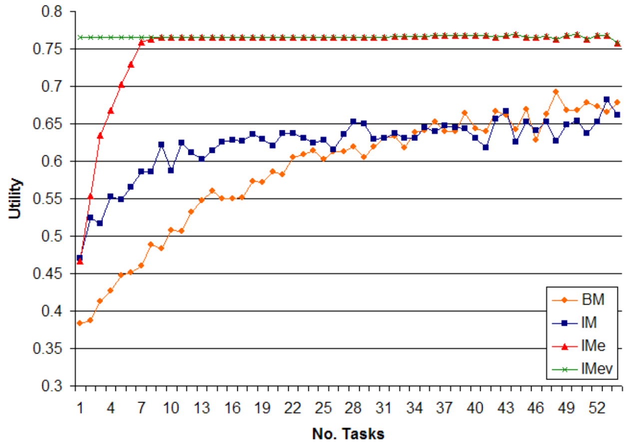

Fig. 2. Utility graphics with different trust models and clustering - evolution of function $\mathcal{U}$ over time

extending role taxonomies. Moreover, those approaches do not take advantages of organisational structures as we do.

There is also much literature about re-organisation in MAS [19, 1], but none of them have included trust mechanisms as a support for re-organising organisational structures. It is the use of trust given by agents what allow the system to re-organise the role taxonomy

Our previous work in [6] also tackled the problem of re-organising organisational structures in virtual organisations of agents, but neither the notion of role - than in current work is much more useful - nor the clustering mechanism are the same. Whilst in this work we use a multidimensional clustering to extend new roles (note that roles are n-dimensional entities - capabilities for any task), in [6] we used a simple bi-dimensional clustering that did not group agents in groups of interactions - as we do - but it specialized single interactions.

6 Conclusions

In this work we have proposed a mechanism for T-MAS that help agents to select counterparts to interact with. This is possible since it endows T-MAS with organisational structures (roles organised in a taxonomy), allowing agents to use trust mechanisms that may take advantage of them. Moreover, role taxonomies evolve over time from past experiences stored in agents' internal structures. We present some results that show that the use of this kind of mechanisms is advantageous for agents in this type of systems.

We are currently working in additional modifications for the mechanism presented in this work to endow it with more functionality. For example, it is necessary to allow agents join and leave from the system. This could be interesting to study since set out the possibility of removing roles when re-organising the system (for example if no agent enacts a specific role anymore). We are also working on different politics to let the system take the decision of re-organising the system on run-time.

References

1. Virginia Dignum, Frank Dignum, and Liz Sonenberg. Towards dynamic reorganization of agent societies. In *In Proceedings of Workshop on Coordination in Emergent Agent Societies*, pages 22–27, 2004.
2. Virginia Dignum, Javier Vázquez-Salceda, and Frank Dignum. Omni: Introducing social structure, norms and ontologies into agent organizations. In *PROMAS*, pages 181–198, 2004.
3. Marc Esteva, Juan A. Rodríguez-Aguilar, Carles Sierra, Pere Garcia, and Josep Lluís Arcos. On the formal specifications of electronic institutions. In *Agent Mediated Electronic Commerce, The European AgentLink Perspective.*, pages 126–147, London, UK, 2001. Springer-Verlag.
4. Prasanna Ganesan, Hector Garcia-Molina, and Jennifer Widom. Exploiting hierarchical domain structure to compute similarity. *ACM Trans. Inf. Syst.*, 21(1):64–93, 2003.
5. Ramón Hermoso, Holger Billhardt, Roberto Centeno, and Sascha Ossowski. Effective use of organisational abstractions for confidence models. In Oguz Dikenelli Gregory O'Hare, Michael O'Grady and Alessandro Ricci, editors, *Engineering Societies in the Agents World*, volume 4457 of *LNAI*, pages 368–383, Berlin, 2007. Springer.
6. Ramón Hermoso, Roberto Centeno, Holger Billhardt, and Sascha Ossowski. Extending virtual organizations to improve trust mechanisms. In *Proceedings of the Seventh International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS 2008*, pages 1489–1492, 2008.
7. http://www.yahoo.com.
8. Jomi Fred Hübner, Jaime Simão Sichman, and Olivier Boissier. Moise+: Towards a structural, functional, and deontic model for MAS organization. In *Proceedings of the First International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS'2002)*, pages 501–502, Bologna, Italy, 2002. ACM Press.
9. T. Dong Huynh, Nicholas R. Jennings, and Nigel R. Shadbolt. FIRE: An integrated trust and reputation model for open multi-agent systems. In *Proceedings of the 16th European Conference on Artificial Intelligence (ECAI)*, 2004.
10. A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: a review. *ACM Comput. Surv.*, 31(3):264–323, 1999.
11. Thomas Juan, Adrian Pearce, and Leon Sterling. Roadmap: Extending the gaia methodology for complex open systems, 2002.
12. J. B. Macqueen. Some methods of classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathemtical Statistics and Probability*, pages 281–297, 1967.
13. Andrea Omicini and Sascha Ossowski. Objective versus subjective coordination in the engineering of agent systems. In Matthias Klusch, Sonia Bergamaschi, Peter Edwards, and Paolo Petta, editors, *Intelligent Information Agents: An AgentLink Perspective*, volume 2586 of *LNAI: State-of-the-Art Survey*, pages 179–202. Springer-Verlag, March 2003.

14. Sarvapali D. Ramchurn, Carles Sierra, Llus God, and Nicholas 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.
15. Jordi Sabater and Carles Sierra. Reputation and social network analysis in multi-agent systems. In *AAMAS '02: Proceedings of the first international joint conference on Autonomous agents and multiagent systems*, pages 475–482, New York, NY, USA, 2002. ACM Press.
16. W. T. Luke Teacy, Jigar Patel, Nicholas R. Jennings, and Michael Luck. Travos: Trust and reputation in the context of inaccurate information sources. *Autonomous Agents and Multi-Agent Systems*, 12(2):183–198, 2006.
17. Bin Yu and Munindar P. Singh. An evidential model of distributed reputation management. In *AAMAS '02: Proceedings of the first international joint conference on Autonomous agents and multiagent systems*, pages 294–301, New York, NY, USA, 2002. ACM Press.
18. Haizheng Zhang, W. Bruce Croft, Brian Levine, and Victor Lesser. A multi-agent approach for peer-to-peer information retrieval. In *In Proceedings of Third International Joint Conference on Autonomous Agents and Multi-Agent Systems*, 2004.
19. Wang Zheng-guang, Liang Xiao-hui, and Zhao Qin-ping. Adaptive mechanisms of organizational structures in multi-agent systems, 2006.