

Organisational Structures in Next-Generation Distributed Systems:

Towards a Technology of Agreement

Authors:

Holger Billhardt¹, Roberto Centeno¹, Carlos E. Cuesta², Alberto Fernández¹,
Ramón Hermoso¹, Rubén Ortiz¹, Sascha Ossowski¹, J. Santiago Pérez-Sotelo¹,
and Matteo Vasirani¹

Affiliation:

¹ Centre for Intelligent Information Technologies (CETINIA)

² Department of Computer Languages and Systems II (DLSI2)

University Rey Juan Carlos

Calle Tulipán s/n, 28933 Móstoles - Madrid, Spain

Abstract. This article provides a brief overview of the field of coordination in multiagent systems, and outlines its relation to current efforts working towards a paradigm for smart, next-generation distributed systems, where coordination is based on the concept of agreement between computational entities. Two examples are provided to visualize the types of mechanisms that can be part of a “technology of agreement”. They explain how techniques from the field of organisations can be used to further coordination and agreement in open multiagent systems. This approach requires autonomy of the coordinated systems and their constituting agents, implying also the need for eventual reorganization

and self-adaptation – hence the article presents also an architectural strategy to tackle the inherent dynamism in these features.

Keywords: multiagent systems, organisational techniques, coordination, agreement technologies, distributed systems, self-adaptation, reorganisation, adaptive architecture.

1 Introduction

Computers and computer networks nowadays mediate an increasing number of transactions and interactions. An appealing way to model and design such applications is by purposefully combining components to which more and more complex tasks can be delegated. These components need to show an adequate level of intelligence, should be capable of sophisticated ways of interacting, and are usually massively distributed, sometimes embedded in all sort of appliances and sensors. In order to allow for an efficient design and implementation of systems of these characteristics, it is necessary to effectively enable, structure, and regulate their communications in different contexts. These characteristics raise some technological challenges. Firstly, the open distributed nature of such systems adds to the *heterogeneity* of its components. The *dynamicity* of the environment calls for a continuous *adaptation* of the structures that regulate the components' interactions, so as to achieve and sustain desired functional properties. On the other hand, non-functional issues related to *scalability*, *security*, and *usability* need to be taken into account. When designing mechanisms that

address these challenges, the notion of *autonomy* becomes central: components may show complex patterns of activity aligned with the different goals of their designers, while it is usually impossible to directly influence their behaviour from the outside.

Coordination in multiagent system (MAS) aims at harmonising the interactions of multiple autonomous components or agents. Therefore, it appears promising to review different conceptual frameworks for MAS coordination, and to analyse the potential and limitations of the work done in that field with regard to some of the aforementioned challenges.

This paper is organised as follows. Section 2 provides a brief overview of coordination in MAS, identifies the notion of *agreement* as a centrepiece of an integrated approach to coordination in open distributed systems, and outlines some research topics related to the vision of a technology of agreement. Two examples are explained, with a certain level of detail, in section 3; they point out how organisational structures could be used to instil coordination and agreement in open multiagent systems, in the realm of matchmaking and trust mechanisms. Moreover, these structures need to be able to evolve and change; in section 4 the potential of such reorganisations is explored, describing the infrastructure which makes them possible and several of the adaptation patterns they provide. Finally, in section 5 some conclusions are drawn.

2 Coordination and Agreement in Multiagent Systems

Originally, agents were conceived as single actors, which had to reconcile a number of conflicting requirements on their inside. But with the advent of MAS, a different method has become possible, causing the focus to change. Now a separate agent can be devoted to a different goal; and the whole system gets the responsibility of reconciling these views and solving the problem. The trade-off continues, but has transformed into a *coordination* problem.

Maybe one of the most accepted definitions of coordination in the MAS field is taken from Organisational Science. It defines coordination as the *management of dependencies* between organisational activities [27]. This definition allows deducing the components of coordination: goals, activities, actors and interdependencies. In summary, when using MAS as a software solution the problem of coordination is always present.

2.1 The Problem of Coordination

In an MAS setting the entities to coordinate are the agents while the coordination objects are usually the goals, actions or plans. Depending on the characteristics of the MAS environment, taxonomy of dependencies can be established, and a set of potential coordination actions assigned to each of them (e.g. [48], [31]). Within this model, the *process* of coordination is to accomplish two major tasks: first, a *detection* of dependencies needs to be performed, and second, a *decision* respecting which coordination action to apply must be taken. A *coordination mechanism* shapes the way that agents perform these tasks [28].

From a *macro-level* (MAS-centric) perspective, the outcome of coordination can be conceived as something global (plan, decision, action etc.). This can be a “shared plan” [39] if the agents reach an explicit agreement during the coordination process, or it just can be the summa of individual plans (or decisions, actions etc. – sometimes called “multi-plan” [32]). At this level, results of coordination can be evaluated as a conjunction of agent goals or taking into account the MAS functionality as a whole. If no such notion can be ascribed to the MAS, other, more basic features can be used instead. A good result of coordination, for instance, often relates to “efficiency”, which frequently comes down to the notion of Pareto-optimality.

The dependency model of coordination appears to be particularly adequate for *representing* relevant features of coordination problems in MAS. Frameworks based on this model have been used to capture coordination requirements in a variety of interesting MAS domains (e.g. [10]). Still, dependency detection may become a rather knowledge intensive task. From a design perspective, coordination is probably best conceived as the effort of *governing the space of interaction* [7] of a MAS, as the basic challenge amounts to how to make agents converge on interaction patterns that adequately (i.e. instrumentally with respect to desired MAS features) solve the dependency detection and decision tasks. A variety of approaches that tackle this problem can be found in [41][29], and [[12].

From the point of view of an agent, the problem of coordination is related to find the sequence of actions that best achieves its goals. In practice, this implies a

series of non-trivial problems. Models of coalition formation determine when and with whom to form a team for the achievement of some common (sub-) goal, and how to distribute the benefits of synergies that arise from this cooperation [43]. Distributed planning approaches [11] determine how to (re-) distribute tasks among team members and how to integrate results. From an individual agent's perspective, the level of trustworthiness of others is central to almost every stage of these processes, so as to determine whether other agents are likely to honour the commitments that have been generated [44].

Several quite different approaches and mechanisms coexist in the field of coordination in MAS [33]. Not all of them are relevant to the challenges for the design of open distributed systems outlined in the introduction. For instance, the whole set of *coupled* coordination mechanisms [47] are effectively useless for the purpose of this paper, as they require having a direct influence on the agent programs. On the other hand, the problem of semantic interoperability is usually outside the scope of MAS coordination models and languages.

2.2 Towards the Agreement between Computational Entities

As already noted, the problem of coordination is always present in an MAS setting and there are several approaches to tackle this problem.

The notion of *agreement* among computational agents appears to be a right concept for the proposal outlined in this paper.

Following a recent research effort in the field of “Agreement Technologies” [1], the process of agreement-based coordination can be conceived based on two main elements:

- (1) *A normative context*, that determines the rules of the game, i.e. interaction patterns and additional restrictions on agent behaviour; and
- (2) *A call-by-agreement interaction method*, where an agreement for action between the agents that respects the normative context is established first; then actual enactment of the action is requested.

Methods and mechanisms from the fields of *semantic alignment*, *norms*, *organization*, *argumentation and negotiation*, as well as *trust and reputation* are envisioned be part of a “sandbox” to build software systems based on a technology of agreement [1].

These can be seen in a “tower” structure, where each level provides functionality and inputs to the one above (see Figure 1):

- *Semantic* technologies should constitute a centrepiece of such an enterprise as semantic problems pervade all the others. The openness in the development of agents, components, or services creates the need for semantic alignment between different ontologies. Every component may have an interface defined according to a (not necessarily shared) ontology [45]. Solutions to semantic mismatches and alignment of ontologies [6] are needed to have a common understanding, e.g. of norms or deals. The use of semantics-based approaches to service discovery and composition will allow exploring the space of possible interactions and, consequently, shaping the set of possible agreements [14].

- At system-level, *norms* are needed to determine constraints that the agreements, and the processes to reach them, have to satisfy. Reasoning about a system's norms is necessary at design-time to assure that the system has adequate properties, but it may also be necessary at run-time, as complex systems usually need dynamic regulations [17]. What set of norms to use in an interaction is a matter of agreement between the entities. These and others considerations require that entities to be highly adaptive to its environment so that agreements including a normative context can be correctly interpreted and executed.
- *Organisational* structures further restrict the way agreements are reached by fixing the social structure of the agents: the capabilities of their roles and the relationships among them (e.g. power, authority) [5]. Many tasks require the recruiting of agents or services to form teams or compound services. These software entities bring in different capabilities that put together may solve a complex task.
- Moving further towards the agent-level, *negotiation* methods are essential to make agents reach agreements that respect the constraints imposed by norms and organisations. Given that the entities are autonomous and black boxes to each other the only way agreements can be reached is via negotiation of its terms and conditions. These methods need to be complemented by an *argumentation*-based approach: by exchanging arguments, the agents' mental states may evolve and, consequently, the status of offers may change [2] [8].
- Finally, agents will need to use *trust* mechanisms that summarise the history of agreements and subsequent agreement executions in order to build long-term

relationships between the agents [46]. Reputation measures are needed to bootstrap the signing of agreements between entities. There are two challenges that need to be addressed to guarantee behaviour: on semantics of the agreements and on social relations between entities [45].

Of course, these methods should not be seen in isolation, as they may well benefit from each other. For instance, in certain situations trust mechanisms may take advantage of the roles structures included in an organisational model, so as to improve their performance when only limited information about previous interactions is available.

3 Organizational Structures and Agreement

This section explains the types of mechanisms that can be part of the agreement technology “sandbox” mentioned previously. In particular, two examples will be provided to explain, with a certain level of detail, how organisational structures could be used to foster coordination and agreement in open MAS will be provided.

Organisational models underlying approaches such as Agent-Group-Role [13], MOISE [22], or RICA [42] provide a rich set of concepts to specify and structure mechanisms that govern agent interactions through the corresponding infrastructures or middleware.

A key notion in most organisational models is the concept of role. The roles can often be organised in a taxonomy, which can be modelled as a pair $\langle R, \leq \rangle$ where R is the set of concepts representing roles and \leq is a partial order among R .

Subsection 3.1 shows how role taxonomies can be used to locate suitable interactions partners, by providing additional information regarding the usability of services in a certain interaction context. Subsection 3.2 outlines how such taxonomies can be used for the bootstrapping of reputation mechanisms, when only limited information about past interactions is available in the system.

3.1 Organisational Structures and Matchmaking Mechanisms

This example refers to service-oriented MAS where the capabilities of agents are modelled in the shape of services that, in turn, are described by some standard service description language. An approach to enriching service descriptions with organisational information will be presented in the following paragraph. For this purpose, simple languages for representing role-based service advertisements and service requests are introduced first.

A service advertisement S is a set of pairs so that

$$S \subseteq \left\{ \langle r, \rho \rangle \mid r \in R, \rho = \bigvee_{i=1}^n \bigwedge_{j=1}^m r_{ij}, r_{ij} \in R \right\}$$

In this definition, r is the role played by the provider in the interaction, and ρ is a set of roles that must be played by the requester agent for the correct accomplishment of the service, given by a formula in disjunctive normal form (DNF).

A service request Q is a set of pairs so that

$$Q = \{ \langle \rho, C \rangle \mid \rho = \bigvee_{i=1}^n \bigwedge_{j=1}^m r_{ij}, r_{ij} \in R, C \subseteq R \}$$

Again, ρ is a DNF role expression (usually atomic) specifying the searched provider roles, and C is a set of roles that define the *capabilities* of the requester (the roles it is able to play).

Although organisational information is not a first-class citizen in service description languages such as OWL-S¹ or WSMO², it is not difficult to incorporate it into them. In OWL-S, for instance, it is possible to include the role description as an additional parameter, called *Service_Roles*, in the case of service descriptions (r and ρ are mapped to *providerRole* and *dependingRoles* tags, respectively), and *Query_Roles* for service requests (ρ and C are mapped to *SearchedProviderRoles* and *CapabilityRoles*) [14].

In many multiagent settings, this kind of organisational information can be used to complement standard I/O based matchmaking in order to improve its performance.

The *semantic match of two roles* r_A (advertisement) and r_Q (query) is made based on the domain ontology R in which they are defined. It is a function that depends on two factors:

- *The Level of match*, which is the (subsumption) relation between the two roles (r_A, r_Q) in the ontology. We differentiate among the four degrees of match proposed by Paolucci et al. [35]: *exact* ($r_A = r_Q$), *plug-in* (r_Q subsumes r_A), *subsumes* (r_A subsumes r_Q) and *fail* (otherwise).
- The *distance* (number of arcs) between r_A and r_Q in the taxonomy (Rada [37]).

We combine both criteria into a final degree of match, which is a real number in the range $[0, 1]$. In this combination, the level of match always has higher priority: the value representing the degree of match is equal to 1 in case of an *exact* match, it varies between 1 and 0.5 in case of a *plug-in* match, rests between 0.5 and 0 in case of a *subsumes* match, and it is equal to 0 in case of a *fail*. These considerations lead to the following equation:

$$dom(r_A, r_Q) = \begin{cases} 1 & \text{if } r_A = r_Q \\ \frac{1}{2} + \frac{1}{2 \cdot e^{\|r_A, r_Q\|}} & \text{if } r_A \text{ is subclass of } r_Q \\ \frac{1}{2} \cdot e^{\|r_A, r_Q\|} & \text{if } r_Q \text{ is subclass of } r_A \\ 0 & \text{otherwise} \end{cases}$$

Where $\|r_A, r_Q\|$ is the distance between r_A and r_Q ($dist(r_A, r_Q)$) in the ontology (if there is a subsumption relation between them). This function guarantees that the value of a *plug-in* match is always greater than the value of a *subsumes* match, and it only considers the distance between the two concepts, rather than the total depth of the ontology tree, which may change depending on the domain. Furthermore, the smaller the distance between concepts (either in the case of *plug-in* or *subsumes* match), the more influence will have a change of distance in the degree of match.

The *semantic match* between a service advertisement S and a query Q (service request) is done by searching the role in S that best matches the one in Q . The degree of match between a role in the request and a service advertisement, given the set of capabilities of the requester, is done by comparing the searched role with every other given role and returns the maximum degree of match. For each

role in the advertisement, the match between the provider roles is made, as well as the match between the depending roles and the capabilities of the requester.

The minimum of both values is considered the degree of match. In case of logical expressions, the minimum is used as combination function for the values in a conjunction and the maximum for disjunctions (which always keep the value resulting of the combination within the range $[0, 1]$).

The following algorithm was developed to determine the degree of match (dom) between a service request (Q) and a service advertisement (S), and constitutes the nucleus of the role-based matchmaker called ROWLS. The implementation relies on the Mindswap Java Library³ for parsing OWL-S service descriptions, and on Jena⁴ for managing OWL ontologies.

Match (Q: service request, S: service advertisement)

dom = 0

FOR ALL CRS_i IN Q.p

dom' = α

FOR ALL r_j IN CRS_i

dom' = min(dom', MatchAtomicRequest(r_j, Q.C, S))

dom = max(dom, dom')

return dom

MatchAtomicRequest(role: Role, Capabilities: SET OF Roles,

S: service advertisement)

dom = 0

```

FOR ALL <r, ρ> IN S {
    dom1 = MatchRole(r,role)
    dom2 = MatchRoleExpr(ρ, Capabilities)
    dom = max(dom, min(dom1,dom2))
return dom
}

MatchRoleExpr(RExpression: SET OF ConjunctiveRoleSet,
              Capabilities: SET OF Roles)
{
    dom = 0
    FOR ALL CRSi IN RExpression {
        dom' = α
        FOR ALL rj IN CRSi {
            dom' = min(dom', MatchRoleInSet(rj, Capabilities))
        }
        dom = max(dom, dom')
    }
    return dom
}

MatchRoleInSet(role: Role, RS: SET OF Roles)
{
    dom = 0
    FOR ALL ri IN RS {
        dom = max(dom, MatchRole(ri,role))
    }
    return dom
}

```

This approach is intended to be complementary to other general-purpose matchmakers. Experiments have been performed combining an implementation of the semantic match between services (ROWLS) with OWLS-MX [26], one of the

leading hybrid matchmakers available to-date. The combination of both matchmakers outperforms a standalone version of the OWLS-MX matchmaker in both efficiency and effectiveness [14].

3.2 Organisational Structures and Trust Mechanisms

In this example an agent shows how can use knowledge about the organisational structure to infer confidence in a situation when no previous experience about a specific interaction is available. Similar to other approaches [23][40], this work sets out from a trust model based on the idea of *confidence* and *reputation*. Both ratings evaluate the trustworthiness of other agents in a particular situation (e.g., playing a particular role in a particular interaction). *Confidence* is a local measure that is only based on an agent's own experiences, while *reputation* is an aggregated value an agent gathers by asking its acquaintances about their opinion regarding the trustworthiness of another agent. Thus *trust* can be defined as a rating resulting from combining *confidence* and *reputation* values.

A typical scenario for the use of a trust model is the following: an agent A wants to evaluate the trustworthiness of some other agent B – playing the role R – in the interaction I . This trustworthiness is denoted as $t_{A \rightarrow \langle B, R, I \rangle} \in [0..1]$, measuring the trust of A in B (playing role R) being a “good” counterpart in the interaction I . When evaluating the trustworthiness on a potential counterpart, an agent can combine its local information (confidence) with the information obtained from other agents regarding the same counterpart (reputation).

Confidence, $c_{A \rightarrow \langle B, R, I \rangle}$, is collected from A 's past interactions with agent B playing role R and performing interactions of type I . The term *Local Interaction Table* (LIT) is used for an agent's data structure storing confidence values for past interactions with any counterpart the agent has interacted with. Each entry corresponds to a *situation*: an *agent* playing a specific *role* in a particular *interaction*. LIT_A denotes agent A 's LIT. An example is shown in Table 1. Each entry in a LIT consists of:

- (i) The Agent/Role/Interaction identifier $\langle X, Y, Z \rangle$,
- (ii) The confidence value for the issue ($c_{A \rightarrow \langle X, Y, Z \rangle}$), and
- (iii) A reliability value ($r_{A \rightarrow \langle X, Y, Z \rangle}$).

The confidence value is obtained from some function that evaluates past experiences on the same situation. It is supposed that $c_{A \rightarrow \langle X, Y, Z \rangle} \in [0..1]$, where higher values represent higher confidence.

Each direct experience of an agent regarding a situation $\langle X, Y, Z \rangle$ changes its confidence value $c_{A \rightarrow \langle X, Y, Z \rangle}$. In this sense, agents have some mechanism to evaluate the behaviour of other agents that they interact with. 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 interaction of type Z . The following formula is used to update confidence:

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

Where $c'_{A \rightarrow \langle X, Y, Z \rangle}$ is the confidence value in A 's LIT before the interaction is performed and $\varepsilon \in [0..1]$ is a parameter specifying the importance given to A 's past confidence value. In general, the aggregated confidence value from past experiences will be more relevant than the evaluations of the most recent interactions.

Reliability ($r_{A \rightarrow \langle X, Y, Z \rangle} \in [0..1]$) measures how certain an agent is about its own confidence in a situation, being based on the work by Huynh et al. [24], taking into account the number of interactions a confidence value is based on and the variability of the individual values across past experiences. Furthermore, it is assumed that $r_{A \rightarrow \langle X, Y, Z \rangle} = 0$ for any tuple $\langle X, Y, Z \rangle$ not belonging to LIT_A .

An agent may build trust directly from its confidence value or it may combine confidence with reputation. This research does not deal with the problem of gathering opinions from other acquaintances by using reputation mechanism, but focus on exploiting past experience to infer expected behaviour on others.

Basic trust models as the one outlined before run into problems when no interactions of a specific type have been performed before and, in addition, social reputation is not available or not reliable. In such a situation, information of the organisational structure can be used to determine an approximate degree of trust.

In particular, one approach consists of using the agent/role confidence $c_{A \rightarrow \langle B, R, _ \rangle}$ (or the agent confidence $c_{A \rightarrow \langle B, _, _ \rangle}$) as an estimation for $c_{A \rightarrow \langle B, R, I \rangle}$ if agent A has no reliable experience about situation $\langle B, R, I \rangle$. This approach relies on the hypothesis that, in general, *agents behave in a similar way in all interactions*

related to the same role. As described before, past experiences are accumulated by agents in form of atomic confidence values for agent/role/interaction tuples in their LIT. This information can be now used to calculate confidence (and trust) values for other organisational elements by accumulating the corresponding entries in an agent LIT.

An agent's trust in a specific role within the organisation is evaluated by the agent/role confidence. It measures the confidence an agent A has in agent B playing a role R and can be calculated by compiling past experiences from any type of interaction where A and B (playing role R) have met:

$$c_{A \rightarrow \langle B, R, _ \rangle} = \frac{\sum_{\langle B, R, _ \rangle_i \in LIT_A} c_{A \rightarrow \langle B, R, _ \rangle_i} \cdot r_{A \rightarrow \langle B, R, _ \rangle_i}}{\sum_{\langle B, R, _ \rangle_i \in LIT_A} r_{A \rightarrow \langle B, R, _ \rangle_i}}$$

The notation $\langle B, R, _ \rangle$ refers to tuples for a fixed agent B and a fixed role R regardless the interaction. Agent/role confidence may be used as an additional evidence measure when calculating $t_{A \rightarrow \langle B, R, I \rangle}$. However, more importantly it provides a manner to evaluate $c_{A \rightarrow \langle B, R, I \rangle}$ (and $t_{A \rightarrow \langle B, R, I \rangle}$) if agent A has none or not enough experience regarding the issue $\langle B, R, I \rangle$, that is, if $r_{A \rightarrow \langle B, R, I \rangle} < \theta$. The importance increases if none of the agents in the organisation has had any experience regarding the issue $\langle B, R, I \rangle$, and therefore, none of the agents could give any (reliable) recommendation. In such a scenario, $c_{A \rightarrow \langle B, R, _ \rangle}$ can provide a valuable approximation of $c_{A \rightarrow \langle B, R, I \rangle}$ for any interaction I.

In a similar way, agents can compute agent confidence $c_{A \rightarrow \langle B, _, _ \rangle}$ – the (global) confidence agent A has in agent B. Agent confidence values can provide a second level of approximation when building $t_{A \rightarrow \langle B, R, I \rangle}$. They may be used as an alternative for $c_{A \rightarrow \langle B, R, I \rangle}$ if there is not even enough expertise for a reliable confidence $c_{A \rightarrow \langle B, R, _ \rangle}$. In a more general environment with agents possibly participating in several organisations, agent confidence may also be used as a gauge to authorise agents to join an organisation.

The previous equation can be adapted to calculate role confidence $c_{A \rightarrow \langle _, R, _ \rangle}$ and interaction confidence $c_{A \rightarrow \langle _, _, I \rangle}$. It is also possible to compute agent/interaction confidence values $c_{A \rightarrow \langle B, _, I \rangle}$; although it is unclear in which settings this measure can be useful.

Role confidence measures an agent's confidence in a specific role within an organisation. This value could be used as default confidence one assigned to agents that just entered an organisation playing a specific role and with no confidence values associated. Interaction confidence provides an estimation of the trust in a concrete interaction within an organisation despite the actual agents that have participated in the interaction. Interaction confidence may be used as a means to choose between several alternative interactions an agent could participate in.

Confidence (and trust) values can also be aggregated for groups of agents – either in general or in relation to one or more interactions or roles. [20]

As an alternative approach to the one presented above lacking confidence values for a situations $\langle B, R, I \rangle$ could be estimated on the basis of considering experiences in similar situations. Based on the assumption that agents behave in a similar way in similar situations, confidence ratings for similar agent/role/interaction tuples can be accumulated to provide evidence for the trustworthiness of the situation $\langle B, R, I \rangle$. Trust can be built by taking into account all the past experiences an agent has, focusing on their degree of similarity between organisational concepts, with the situation of interest $\langle B, R, I \rangle$. In particular, trust can be calculated as a weighted mean over all the confidence values an agent has accumulated in its LIT. This is shown in the following equation:

$$t_{A \rightarrow \langle B, R, I \rangle} = \frac{\sum_{\langle X, Y, Z \rangle \in LIT_A} c_{A \rightarrow \langle X, Y, Z \rangle} \cdot \omega_{A \rightarrow \langle X, Y, Z \rangle}}{\sum_{\langle X, Y, Z \rangle} \omega_{A \rightarrow \langle X, Y, Z \rangle}}$$

$\omega_{A \rightarrow \langle X, Y, Z \rangle}$ 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 issue $\langle B, R, I \rangle$ in the following way:

$$\omega_{A \rightarrow \langle X, Y, Z \rangle} = r_{A \rightarrow \langle X, Y, Z \rangle} \cdot \text{sim}(\langle X, Y, Z \rangle, \langle B, R, I \rangle)$$

The similarity function $\text{sim}(\langle X, Y, Z \rangle, \langle B, R, I \rangle)$ is computed as the weighted sum of the similarities of the individual elements (agent, role and interaction) as it is shown in the following equation:

$$sim(\langle X, Y, Z \rangle, \langle B, R, I \rangle) = \begin{cases} \beta \cdot sim_R(R, Y) + \gamma \cdot sim_I(I, Z) & \text{if } X = B \\ 0 & \text{otherwise} \end{cases}$$

Where $sim_R(R, Y), sim_I(I, Z) \in [0..1]$ measures the similarity between roles and interactions, respectively, and β and γ with $\beta + \gamma = 1$, are parameters specifying the sensibility regarding the individual similarities.

Role similarities can be inferred from role taxonomies contained in an organisational model. In particular, $sim_R(R, R')$ can rely on a *distance function* that estimates the similarity between two roles on the basis of their proximity in the taxonomy. The same holds for $sim_I(I, I')$ when interaction taxonomy is available in an organisational model [21].

Especially if an agent has no reliable experience about a particular agent/role/interaction situation, the organisation-based approach explained can be used to estimate trust without the necessity to rely on the opinions of other agents. So, role and interaction taxonomies can help making agents that use trust mechanisms less vulnerable to dishonest counterparts, as there is less need to rely on third-party information.

4 Adaptation and Self-Organisation

In previous sections, organisational structures and mechanisms have been presented to explain how they could be used to foster coordination and agreement in MAS environments. The growing complexity of software is emphasizing the need for systems that have autonomy, robustness and adaptability among their most important features. It is accepted nowadays that MAS have been developed

as a generic approach to solve complex problems. However, in order to fulfil their promise of generality and extensibility, they should also reach self-adaptivity, i.e. the capability of autonomously adapting to changing conditions. This feature requires agents to be able to alter their own configuration, and even their own composition and typing. Their reorganisations can be seen, therefore, as the first necessary steps to reach actual self-adaptivity.

Beyond previous concepts, this section proposes an architectural solution to tackle the dynamism, which will be supported by an emergent agreement - an evolving architectural structure based on combining predefined controls and protocols. These are handled in the context of a service-oriented, agent-based and organisation-centric framework. Next subsections discuss not only the architectural framework but also the mechanisms to change their composition patterns and element types, which are necessary to achieve real self-adaptivity.

4.1 The Basic Framework for Self-Organisation

As the proposed approach is based on service-oriented concepts, the main idea is to export the agent system as a system of services, and the environment must be truly adaptive and dynamic, it requires the use of rich semantic and highly technological capabilities. Therefore, it is considered a wise use of *agents* in a broader context, with an upper layer of services added to provide, in particular, the *interoperability* feature. It is easy to conceive a service to present the operational capabilities of an agent or a collection of agents as an organisation, which in turn provides services. Using agents allows the explicit treatment of

semantics, a structured coordination, the use of a methodology to service development, to structure them into organisations, and the use of their learning capacity, among others features.

Implicit in the definition of MAS is the need to *register* agents in the system, to separate those ones who belong to the architecture from those who do not. The same approach will be used to identify services. To allow their external access, they will be explicitly registered and grouped as part of a service.

The current research, which is included as part of the OVAMAH project [34], is extending the objectives of the original platform THOMAS [5]. Besides providing the necessary technology for the development of virtual organisations in open environments, it will allow to facilitate dynamic answers for changing situations by means of the adaptation and/or evolution of the organisations. For example, agents forming an organisational unit could create (or remove) another unit, affecting the groups of the system; decide the moment to add or delete norms; the social relationship between roles could change at runtime, the conditions to activate/deactivate, as well as the cardinality of roles; the system topology (given by the relationships) could be changed also at runtime and then validate the changes with objectives and organisational type; the services could be matched to new roles; etc.

The framework is evolving (currently adapting to OSGi [30] specification) and the applications are modularizing into smaller entities called bundles. These entities can be installed, updated or removed on the fly and dynamically, provide the ability to change the system behaviour without ever having to disrupt its

operation. Among the services provided by this standard, the Service Tracker appears as particularly relevant, in the light of the proposed approach. This service makes possible to track other registered services on the platform. It is used to ensure that the services to be provided are still available or not.

In summary, the evolution of the agreement-based approach, including the concepts and constructs that it describes, has already shown its relevance. The main concern now, beyond performance issues, is the essential dynamism and the adaptive functionality required by the underlying architecture.

4.2 Adaptive Organisations Based on Initiatives

A group of individuals can be arranged into certain structures, depending on concrete goals, and they can be formed by using two different kinds of mechanisms: *controls* and *protocols*, which are both based in limiting the range of available actions. The formers can be seen as elements that either enforce or forbid specific interactions (or architectural connections). Self-adaptive structures, being typically centralized [2], show many classic examples of this kind: most of them manifest explicit control loops, inspired in *regulators* of classic control theory. On the other hand, protocols, which either enable or channel behaviour, are based on consensus and agreements. They can be described generically as the way to control decentralized (even distributed) structures [18]. Basically, when protocols are present, every agent knows the way to interact with the rest; it is necessary to comply with them to be able to communicate, but at the same time they are also *regulating* the development of the interacting structure itself.

These two mechanisms define a wide spectrum of regulation, in which agent organisations and their architectures are simultaneously harnessed by atomic, unary *controls* (norms, limits, locks, control loops or constraints) and multiple, connective *protocols* (hubs, bridges, channels, or spaces). It is important to note that the purpose of these mechanisms is to “discover” a suitable structure of controls and protocols so that a global structure can emerge. These elements make possible to define the main inner structures in order to obtain agreement-based organisations. Once a primary structure can be defined, an elemental group emerges as a preliminary organisation, which will be referred as *an initiative*: not yet fully established, but still evolving.

Nevertheless, the *initiative* can continue growing and mutating because of its *adaptive* nature, but when it has some “stable” structure, it can be called *organisation*. This “stable” structure is achieved when all the participants can afford the necessary *agreement* in order to gain the objective. This process can be thought as the system moving to a new state, in which the structure of the “past” is supplanted by a “new” emergent structure. Obviously, this novel structure admits new elements because of the dynamic environment, but now one of its goals is to reinforce its nature.

An *initiative* can be generated from patterns, named *adaptation patterns*, where the term is used in an architectural sense. They are pre-designed from the required services of an *initiative* and the corresponding semantic refining. Some of them have been already identified, and receive such names as *Façade*, *Mediator*, or *Surveyor*, among others (see Subsection 4.3). The patterns represent a fragment of

a static structure, leading to a dynamic one, the *initiative*, reaching a “stable” form, the organisation.

4.3 Adaptation Patterns

As already noted, the *adaptation patterns* are pre-designed from the required services of an *initiative* and for the corresponding semantic refinement. Particularly, these are not classic object-oriented patterns, because they are defined in a different context: they are architectural patterns.

According to [38] it is possible to classify the architectural design patterns as follows: monitoring (M), decision-making (DM), or reconfiguration (R) based on their objective. M and DM patterns can also be classified as either creational (C) or structural (S), as defined in [19]. Likewise, R patterns can also be classified as behavioural (B) and structural (S) since they specify how to physically restructure an architecture once the system has reached a quiescent or safe state for adaptation. Several of these patterns have been already identified for the proposed approach. In Table 2, for instance, three of them are described: *Façade*, *Mediator*, and *Surveyor*.

Obviously, there are more patterns and not all of them describe only roles. For instance, the *Surveyor Election* defines the protocol (one among many) to decide the next *surveyor*; and *Surveyor Change* describes a protocol to demote the current *surveyor* and forward its knowledge to a new one.

All these pre-figured changes are applied to organisations that have reached a *quiescent* or safe state for adaptation [25]. In this case, namely *pure adaptation*,

the importance lies in the way that an existing organisation has to adapt to a new behaviour. First, it has to realize that a change has occurred, i. e. a change can emerge in an intrinsic way [36], and then it has to adapt itself.

There are several scenarios to develop this adaptive behaviour (typically four variants each), reaching ultimately an “stable” configuration for an initiative, which therefore becomes an *organisation*.

As already noted, the system is ultimately conceived as a *service-oriented architecture*; so methodologically, the first stable organisations must be conceived as the providers for certain high-level services. Then, these services must be proposed as the starting point for the functional definition of those first organisations.

5 Discussion

An overview of different approaches to coordination in the MAS field has been presented in this article. It has been argued that the *notion of agreement is essential to instil coordination in open distributed systems*. Some existing technologies from the field of MAS coordination can be applied to this respect, and others – semantic technologies, in particular – need to be added. To explain the types of mechanisms that can be part of a “technology of agreement”, two examples with a certain level of detail have been provided to show how techniques from the field of organisations can be used to further coordination and agreement in open MAS.

It was pointed out how organisational structures can be used to complement traditional matchmaking mechanisms in order to enhance their performance. A role-based matchmaking component has been implemented and evaluated in combination with the well-known OWLS-MX matchmaker. Furthermore, it has been argued that organisational structures can be used to improve reputation mechanisms in situations where only a limited amount of information regarding previous interactions is available. Current work focuses on how, in turn, the history of interactions can be used to evolve organisational structures [20], [21].

These structural concepts have also been explored as the basis of an architectural approach to provide self-adaptivity to agent systems. The proposed concept of *initiative* must be considered as a starting point to provide mechanisms to change the composition patterns and element types within such systems.

The required dynamism can be supported by an *emergent* agreement - an evolving architectural structure, based on combining predefined *controls* and *protocols*. These mechanisms are handled in the context of the service-oriented, agent-based and organisation-centric framework defined in AT, which is also compatible to the OSGi standard. The key idea is to create an architectural context, in which agents are *coordinated* and *reorganised* by inclusion in preliminary structures – i.e. adaptation patterns– and then in stable organisations. This approach seems promising, even when its more important consequences are yet to be studied.

The concept of *agreement* is still evolving and the process of defining its limits continues, but even at this stage, the approach have already proven its utility and expressive power.

Several currently research efforts, which are being carried out as European projects, may contribute to the development of a “technology of agreement” [9]. These efforts promote the emergence of a new paradigm for next-generation distributed systems based on the agreement in a society of computational agents.

Acknowledgements

Some ideas reported in this article draw upon joint work with our partners in the “Agreement Technologies” project. This work has been partially funded by the Spanish Ministry of Science and Innovation through grants TIN2006-14630-C03-02, TIN2009-13839-C03-02, MULTIPLE (TIN2009-13838), and CSD2007-0022 (CONSOLIDER, INGENIO 2010).

References

- [1] Agreement Technologies project homepage: www.agreement-technologies.org/
- [2] Amgoud, L.; Dimopolous, Y.; Moraitis, P. (2007): A unified and general framework for argumentation-based negotiation. Proc. 6th Int. Joint Conference on Autonomous Agents and MultiAgents Systems, (AAMAS'2007). IFAAMAS, pages 963-970
- [3] Andersson, J.; de Lemos, R.; Malek, S.; and Weyns, D. Modeling Dimensions of Self-Adaptive Software Systems. In Software Engineering for Self-Adaptive Systems, LNCS, vol. 5525:27-47, Springer, (2009).
- [4] Argente, E., Botti, V., Carrascosa, C., Giret, A., Julian, V., and Rebollo, M.: An Abstract Architecture for Virtual Organisations: The THOMAS Project. Technical report, DSIC, Universidad Politécnica de Valencia (2008).
- [5] Argente, E.; Julian, V.; and Botti, V. (2006): MultiAgent System Development based on Organizations. Elec. Notes in Theoretical Computer Science Vol. 150 No. 3, pages 55-71
- [6] Atienza, M.; Schorlemmer, M. (2008): I-SSA – Interaction-situated Semantic Alignment. To appear in: Proc. Int. Conf. on Cooperative Information Systems (CoopIS-2008)
- [7] Busi, N.; Ciancarini, P.; Gorrieri, R.; Zavattaro, G. (2001): Coordination Models – A Guided Tour. Coordination of Internet Agents: Models,

- Technologies, and Applications (Omicini et al., eds.). Springer-Verlag, pages 6–24
- [8] Caminada, M.; Amgoud, L. (2007): On the evaluation of argumentation formalisms. *Artificial Intelligence Journal*, V.171 (5-6). Elsevier, pages 286-310.
 - [9] COST Action IC0801 homepage: www.agreement-technologies.eu
 - [10] Decker, K. (1996). TAEMS: A Framework for Environment Centered Analysis and Design of Coordination Mechanisms. *Foundations of Distributed Artificial Intelligence* (O'Hare and Jennings, eds.), John Wiley and Sons.
 - [11] Durfee, E. (2001): Distributed Problem Solving and Planning. *Multiagent Systems and Applications* (Luck et al., eds.), LNAI 2086. Springer, pages 118-149
 - [12] Esteva, M; Rosell, B.; Rodríguez-Aguilar, J.A.; Arcos, J. L. (2004): AMELI – An agent-based middleware for electronic institutions. *Proc. Int. Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-2004)*. ACM Press, pages 236-243
 - [13] Ferber, J.; Gutknecht, O.; Fabien, M. (2003): From Agents to Organizations – An Organizational View of Multiagent Systems. *Agent-Oriented Software Engineering IV* (Giorgini, Müller, and Odell, eds.), LNCS 2935. Springer, pages 214-230
 - [14] Fernández, A.; Ossowski, S. (2008). Exploiting Organisational Information for Service Coordination in Multiagent Systems. *Proc. of the*

- Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS-2008). IFAAMAS, pages 257-264
- [15] Fernández, A.; Polleres, A.; Ossowski, S. (2007). Towards Fine-grained Service Matchmaking by Using Concept Similarity. Workshop on Service Matchmaking and Resource Retrieval in the Semantic Web (SMR2) (Di Noia et al., eds.). CEUR Workshop Proceedings Vol. 243, pages 31-34
- [16] FIPA Abstract Architecture Specification. Technical Report SC00001L, Foundation for Intelligent Physical Agents. FIPA TC Architecture (2002).
- [17] Gaertner, D.; García-Camino, A.; Noriega, P.; Rodríguez-Aguilar, J.A.; Vasconcelos, W. (2007): Distributed norm management in regulated multiagent systems. Proc. Int. Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-2007). IFAAMAS, pages 624-631
- [18] Galloway, A.R. Protocol: How Control Exists after Decentralization. MIT Press, 2004.
- [19] Gamma, E., Helm, R., Johnson, R., and Vlissides, J.: Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley Professional (1994).
- [20] Hermoso, R.; Billhardt, H.; Ossowski, S. (2007): Integrating Trust in Virtual Organisations. Proc. COIN'06, Workshop in Coordination, Organisation, Institutions and Norms in Multiagent Systems (AAMAS-2006). LNAI 4386, pages 17–29
- [21] Hermoso, R.; Centeno, R.; Billhardt, H.; Ossowski, S. (2008): Extending Virtual Organizations to improve trust mechanisms (Short Paper). Proc.

- Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS-2008). IFAAMAS, pages 1489-1492
- [22] Hubner, J.; Sichman, J.; Boissier, O. (2006): Developing organised multiagent systems using the MOISE+ model: programming issues at the system and agent levels. *Int. Journal of Agent-Oriented Software Engineering*, Vol.1, No.3/4. Inderscience, pages 370-395
- [23] Huynh, T.D.; Jennings, N.R.; Shadbolt N. (2004): FIRE: An integrated trust and reputation model for open multiagent systems. In *Proceedings of the 16th European Conference on Artificial Intelligence (ECAI)*, pages 18-22
- [24] Huynh, T.D.; Jennings, N.R.; Shadbolt N.R. (2004): Developing an integrated trust and reputation model for open multiagent systems. In *Proceedings of 7th International Workshop on Trust in Agent Societies (AAMAS)*, pages 65-74
- [25] Kramer J.; and Magee J. (1990): The Evolving Philosophers Problem: Dynamic Change Management, *IEEE Trans. on Software Eng.*, 16 (11), pages 1293-1306
- [26] Klusch, M.; Fries, B.; Sycara, K. (2006): Automated Semantic Web Service Discovery with OWLS-MX. *Proc. Int. Conf. on Autonomous Agents and MultiAgent Systems (AAMAS-2006)*. ACM Press, pages 915-922
- [27] Malone, T.; Crowston, K. (1994): The Interdisciplinary Study of Coordination. *Computing Surveys* 26 (1). ACM Press, pages 87-119

- [28] Omicini, A.; Ossowski, S. (2003): Objective versus Subjective Coordination in the Engineering of Agent Systems. Intelligent Information Agents – The European AgentLink Perspective (Klusck et al, eds). Springer-Verlag, pages 179–202
- [29] Omicini, A.; Ossowski, S.; Ricci, A. (2004): Coordination Infrastructures in the Engineering of Multiagent Systems. Methodologies and software engineering for agent systems – The Agent-Oriented Software Engineering Handbook (Bergenti, Gleizes, and Zambonelli, eds.). Kluwer, pages 273-296
- [30] OSGi: formerly known as the Open Services Gateway initiative.
<http://www.osgi.org/>
- [31] Ossowski, S. (1999): Co-ordination in Artificial Agent Societies, LNAI 1535. Springer.
- [32] Ossowski, S. (2001): Constraint Based Coordination of Autonomous Agents. Electronic Notes in Theoretical Computer Science 48. Elsevier, pages 211-226
- [33] Ossowski, S.; Menezes, R. (2006): On Coordination and its Significance to Distributed and MultiAgent Systems. Journal of Concurrency and Computation – Practice and Experience 18(4). Wiley, pages 359-370
- [34] OVAMAH - Organizaciones Virtuales Adaptativas: Técnicas y Mecanismos de Descripción y Adaptación.
<http://gti-ia.dsic.upv.es:8080/ovamah>

- [35] Paolucci, M.; Kawamura, T.; Payne, T.; and Sycara, K. (2002): Semantic matching of web services capabilities. In Proceedings of the First International Semantic Web Conference on The Semantic Web, Springer-Verlag, pages 333-34
- [36] Prokopenko, M., Boschetti, F., and Ryan, A.J. (2008): An Information-Theoretic Primer on Complexity, Self-Organisation, and Emergence. Complexity, Vol.15-1:11-28. Wiley Periodicals, Inc.
- [37] Rada, R.; Mili, H.; Bicknell, E.; Blettner, M. (1989): Development and application of a metric on semantic nets. IEEE Trans. on Systems, Man and Cybernetics 19(1), pages 17–30
- [38] Ramírez, A. J., and Cheng, B. H. (2010): Design Patterns for Developing Dynamically Adaptive Systems. ICSE2010-SEAMS, pages 49-58
- [39] Rosenschein, J.; Zlotkin, G. (1995): Designing Conventions for Automated Negotiation. AI Magazine 15 (3), pages 29–46
- [40] Sabater, J.; Sierra, C. (2001): REGRET: a reputation model for gregarious societies. In proceedings of the Fifth International Conference on Autonomous Agents (AGENTS). ACM Press, pages 194-195
- [41] Schumacher, M.; Ossowski, S. (2006): The governing environment. Environments for Multiagent Systems II (Weyns et al., eds.). LNCS 3830. Springer, pages 88-104
- [42] Serrano, J.M.; Ossowski, S. (2004): On the Impact of Agent Communication Languages on the Implementation of Agent Systems.

- Cooperative Information Agents VIII (Klusch et al., eds.). LNCS 2782. Springer, pages 92–106
- [43] Shehory, O., Sycara, K. y Somesh, J. (1998): Multiagent Coordination through Coalition Formation. Intelligent Agents IV – Agent Theories, Architectures and Languages (Singh, Rao, and Wooldridge). LNCS 1365. Springer, pages 143-154
- [44] Sabater, J.; Sierra, C. (2005): Review on Computational Trust and Reputation Models. Artificial. Intelligence Review 24(1). Springer, pages 33-60
- [45] Sierra, C.; Botti, V.; Ossowski, S. (2011): Agreement Computing. In: KI - Künstliche Intelligenz Vol 25 (1), pages 57-61
- [46] Sierra, C.; Debenham, J. (2007): Information-Based Agency. Proc Int. Joint Conference on AI (IJCAI-2007). AAAI Press, pages 1513-1518
- [47] Tolksdorf, R. (2000): Models of Coordination. Engineering Societies in an Agent World (Omicini et al., eds.). Springer-Verlag
- [48] Von Martial, F. (1992): Co-ordinating Plans of Autonomous Agents. LNAI 610, Springer

Authors' biographical notes.

Holger Billhardt obtained his BSc and Masters degree at the Technical University of Leipzig (Germany) in 1994 and his PhD in Computer Science at the Technical University of Madrid (Spain) in 2003, where he was working as a research fellow between 1997 and 2001. Currently he is an Associated Professor at the University Rey Juan Carlos in Madrid (Spain), where he is member of the Centre for Intelligent Information Technologies (CETINIA). His research is concerned with multiagent systems; in particular, regarding regulation and coordination mechanisms for distributed dynamic and open environments. He has published more than 40 papers in international journals and conferences.

Roberto Centeno is a PhD student and a member of the Centre for Intelligent Information Technologies (CETINIA) at University Rey Juan Carlos in Madrid (Spain). Currently he is working as Assistant Professor at the Computer Languages and Systems Department at UNED (Spain). He obtained his computer science degree at University Rey Juan Carlos in 2006. His main research interests focus on multiagent systems, virtual organizations, adaptive mechanisms and trust and reputation. He has published several research papers at international conferences and participated in several research projects focused on the application of Artificial Intelligence techniques to real world problems.

Carlos E. Cuesta is an Associate Professor of Software Engineering, and the Associate Dean for Academic Affairs in the School of Computer Science & Engineering at Rey Juan Carlos University in Madrid (Spain). He obtained his B.Sc. (1994) and Master's (1997) degrees at the University of Valladolid (Spain), where he also received his Ph.D. (2002). His primary research interest is software architecture, unfolding also in topics such as coordination, invasive composition, formal methods, service-oriented and model-driven architectures, agent-oriented technologies, architectural design knowledge, or self-adaptation. He has published several papers in international conferences and journals, and is a member of the steering committee of the ECSA series of conferences.

Alberto Fernández received the PhD degree from the University Rey Juan Carlos, Madrid (Spain) in 2007, where he is an Associated Professor. He is also member of the Artificial Intelligence Group and the Centre for Intelligent Information Technologies (CETINIA). His research interests include coordination in multiagent systems, semantic web, semantic web services and service oriented MAS. He has published more than 40 papers in international journals and conferences, and has participated in more than 20 research projects. He is a reviewer in several international conferences and journals.

Ramón Hermoso is a PhD student and member of the Centre for Intelligent Information Technologies (CETINIA) at the University Rey Juan Carlos in Madrid (Spain). He obtained his MSc degree in Informatics at the University Rey

Juan Carlos in 2004. His research interests are focused on the field of Artificial Intelligence, particularly on multiagent systems and the design of social mechanisms in agent societies. He is author of several research papers in different books and international conferences. He is program committee of several international workshops and conferences.

Rubén Ortiz is Teaching Assistant in computer science at the University Rey Juan Carlos in Madrid (Spain), where he is also is a PhD student and member of the Centre for Intelligent Information Technologies (CETINIA). He received his MSc degree in Computer Science from the University Rey Juan Carlos. Since 2000 his research has focused on several areas related to design multiagent systems and their applications.

Sascha Ossowski is Full Professor of Computer Science and Director of the Centre for Intelligent Information Technologies (CETINIA) at University Rey Juan Carlos in Madrid (Spain). He obtained his MSc degree in Informatics from University of Oldenburg in 1993, and received a PhD in Artificial Intelligence from Technical University of Madrid in 1997. His research focuses on the application of distributed AI techniques to real world problems. He is chair of the COST Action on Agreement Technologies, chairs the Board of Directors of the European Association for Multiagent Systems, and is a member of the steering committee of the ACM SAC conference series.

J. Santiago Pérez-Sotelo is a PhD student and member of the Centre for Intelligent Information Technologies (CETINIA) at the University Rey Juan Carlos in Madrid (Spain). He received his degree as Information Systems Engineer from the Universidad Tecnológica Nacional - Facultad Regional Resistencia (Argentina) in 2002, where he has worked until 2007 in the Artificial Intelligence area. He obtained his MSc in Information Technologies and Information Systems in 2009 from the University Rey Juan Carlos, where he is also working since 2008. His research interests include intelligent agents, software architecture and service-oriented systems. He has published several research papers in international conferences.

Matteo Vasirani is Teaching Assistant and Associate Director for Training and Promotion of the Centre for Intelligent Information Technologies (CETINIA) at University Rey Juan Carlos in Madrid (Spain). He obtained his MSc degree in Informatics from the University of Modena (Italy) in 2004, and his PhD in Artificial Intelligence from University Rey Juan Carlos in 2009. His research interests span from coordination and learning in multiagent systems, to intelligent infrastructures for transportation and power systems. He is author of several publications in journals, books and international conferences, and he has participated in more than 10 research projects, funded by the European Commission and the Spanish Government.

Table 1: LIT (Local Interaction Table)

It is an agent's data structure storing confidence values for past interactions with any counterpart the agent has interacted with.

$\langle X, Y, Z \rangle$	$c_{A \rightarrow \langle X, Y, Z \rangle}$	$r_{A \rightarrow \langle X, Y, Z \rangle}$
$\langle a_9, r_2, i_3 \rangle$	0.2	0.75
$\langle a_2, r_7, i_1 \rangle$	0.7	0.3
\vdots	\vdots	\vdots
$\langle a_9, r_2, i_5 \rangle$	0.3	0.5

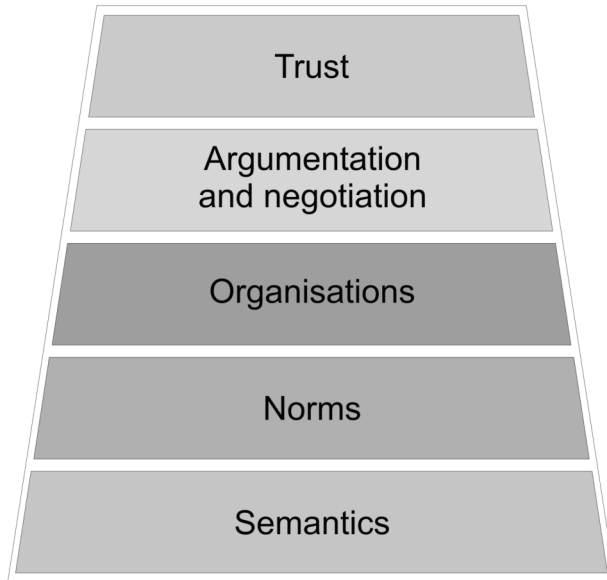
Each entry corresponds to a situation: an agent playing a specific role in a particular interaction. LIT_A denotes agent A's LIT.

Table 2: Adaptation Patterns: architectural design patterns.

Name	Category	Description
<i>Façade</i>	M, S	To be able to easily interact with an organisation, which still lacks a defined structure, some agent has to represent the organization itself in terms of interaction. This agent redirects any incoming communication.
<i>Mediator</i>	R, B	During the emergence process, the organization is not yet established, and data services are probably not working. Some agent must act as a <i>mediator</i> , which makes possible to access to data sources, although indirectly, and also to perform the necessary (<i>semantic</i>) translations.
<i>Surveyor</i>	R, S	During the emergence process, at least one agent must monitor the growing of the initiative itself, both to decide when new elements are inserted, and also when the initiative forms a “stable” organization. It has access to the pattern library and decides when a certain pattern must be triggered.

Figure 1: Agreement Technologies' original tower (layered) structure [1].

Figure 1:



Footnotes

- 1 - <http://www.daml.org/services/owl-s>
- 2 - <http://www.wsmo.org>
- 3 - <http://www.mindswap.org/mhgrove/kowari>
- 4 - <http://jena.sourceforge.net>