Exploiting Organisational Information for Service Coordination in Multiagent Systems

Alberto Fernández
Universidad Rey Juan Carlos
Tulipán s/n, 28933
Móstoles (Madrid) - Spain
+34 91 488 7084
alberto.fernandez@urjc.es

Sascha Ossowski
Universidad Rey Juan Carlos
Tulipán s/n, 28933
Móstoles (Madrid) - Spain
+34 91 664 7485
sascha.ossowski@urjc.es

ABSTRACT
Service-Oriented Computing and Agent Technology are nowadays two of the most active research fields in distributed and open systems. However, when trying to bridge the two worlds, it becomes apparent that the interaction-centric approach of multiagent systems may affect the way services are modelled and enacted, and vice versa. We claim that organisational models that underlie multiagent interactions are crucial in order to take advantage of this interrelation.

In this paper we present an approach for modelling organisational structures in service-oriented multiagent systems, and show how it affects semantic service descriptions. We also present approaches to service matchmaking as well as to service composition, capable of exploiting organisational information in service descriptions. In both cases, we prove experimentally the validity of our approach.

Categories and Subject Descriptors
I.2.m [**Artificial Intelligence**]: Miscellaneous

General Terms
Algorithms, Design.

Keywords
Semantic Web Services, Multiagent systems, Organisational concepts, Service description, Service composition.

1. INTRODUCTION
Service-Oriented Computing (SOC) [7] is a novel computing paradigm that conceives services as basic elements to develop applications and systems. Services are self-describing, platform-independent computational entities that can be described, published, discovered, orchestrated and invoked by other software entities in order to support rapid and low-cost composition of distributed applications.

Agent Technology provides designers with an interaction-centric way of designing open and distributed software systems [12]. There is a growing awareness that organisational models are fundamental to regulate open multi-agent systems and to promote coordination among agents so as to instil desired properties [15][17].

Although the two research fields have different backgrounds and motivations, there is a growing interest in bridging the two worlds: on the one hand, software agents can be viewed as potential users and providers of semantic web services, on the other, web service technology can be used to support the interactions in multiagent systems [4]. To this respect, the notion of Service-Oriented Multi-Agent Systems (SOMAS) has recently been put forward [8].

However, the integration of both worlds in SOMAS is usually still quite shallow, as it frequently refers to the conceptual level, but rarely affects the proper mechanisms used in the system: agents are often seen as wrappers of services that do not concern the functioning of the service itself. In this paper we show how key concepts from the multiagent world can be used to extend and improve essential mechanisms of service-orientation. In particular, we show how organisational models can be incorporated into service descriptions, so as to improve service coordination mechanisms in SOMAS.

The paper is organized as follows: In Section 2 we outline our approach for modelling and representing organisational information with regard to SOMAS services. Section 3 presents a matchmaking mechanism that exploits organisational information to improve the dynamic discovery of relevant services. Section 4 puts forward a filtering approach to semantic service composition that draws upon organisational information. In Section 5 we present our conclusions and point to future lines of work

2. SOMAS SERVICE DESCRIPTIONS
In this section we first outline our notion of organisational information in SOMAS. We then show how relevant parts of it can be incorporated into service advertisements and requests, and represented by OWL-S service descriptions.

2.1 SOMAS Organisational Model
Our SOMAS organisational model extends part of the RICA meta-model described in [15]. Figure 1 shows part of this extension which is relevant to this paper.

Services are provided by *agents*, which are able to engage in different *types of interactions* when executing a service by playing the *roles* that participate in them. Services can be simple (in which case they are provided by one agent) or can be

Cite as: Exploiting Organisational Information for Service Coordination in Multiagent Systems, A. Fernández, S. Ossowski, *Proc. of 7th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2008)*, Padgham, Parkes, Müller and Parsons (eds.), May, 12-16., 2008, Estoril, Portugal, pp. 257-264.

orchestrated into a *composite service* to provide a more complex functionality that no single agent is able to offer. In this case, a *team* of agents collaborate by enacting parts of the whole composite service.

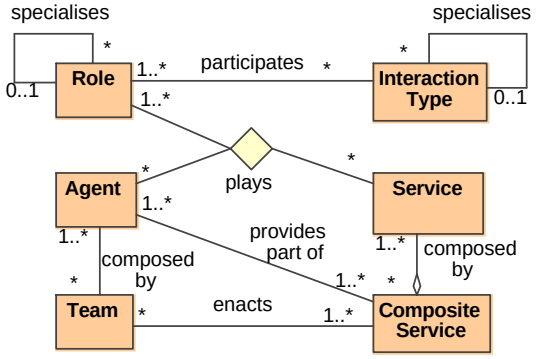

Figure 1 – SOMAS Organisational Meta-Model

In this paper, we will focus exclusively on organisational information provided by roles and interactions as shown in the class diagram of Figure 1. This information can be obtained by applying the RICA design method for multiagent systems [15] to a SOMAS for a particular domain. In [3] we show in detail how, setting out from a set of complex use cases in the field of medical emergency assistance, an ontology can be derived that contains a taxonomy of types of interactions and a taxonomy of roles that take part in those interactions.

Definition 1. A SOMAS roles and interactions ontology is a quadruple $O = <R, I, \leq, \Delta>$ where:

- R is the set of concepts representing roles
- I is the set of concepts representing interactions
- $\leq$ is a partial order among R and among I representing a subclass relation among roles and types of interactions
- $\Delta: R \rightarrow I$, an order-preserving function that associates every role with the type of interaction in which it participates.

For instance, the use cases about medical emergency assistance scenario described in [3] include subdialogues regarding a second opinion service, where interactions about medical advisement, medical information exchange and medical explanation take place. A medical advisement interaction (*ma*) can be seen as a specialisation of general advisement interactions (*a*), i.e: $a, ma \in I$, $ma \leq a$. Furthermore, a medical advisor (*mar*), subclass of a generic advisor (*ar*) plays one of the roles in this interaction ($ar, mar \in R$; $mar \leq ar$; $\Delta(mar) = ma$, $\Delta(ar) = a$). Analogously, a medical information exchange (*mi*) is a kind of a general information exchange ($i, mi \in I$, $mi \leq i$; $ir, mir \in R$, $mir \leq ir$; $\Delta(mir) = mi$, $\Delta(ir) = i$), and medical explanation (*me*) specialises the explanation interaction ($e, me \in I$, $me \leq e$; $er, mer \in R$, $mer \leq er$; $\Delta(mer) = me$, $\Delta(er) = e$). Note that part of the ontology is generic and can be reused in other domains (e.g. financial advisor is another specialisation of the generic role advisor, $ar, far \in R$; $far \leq ar$ …).

In [15], an analysis of FIPA ACL has lead to a minimal, domain independent taxonomy of roles and interactions that can be used as a basis for domain specific extensions like the one sketched above. Please notice that the invocation of a service in a standard SOC (i.e. without any notion of an underlying organisational model) would lead to a system with to just one type of interaction (*request*), playing the service provider the *requestee* role.

2.2 Role-Based Service Descriptions

In SOC, two fundamental functionalities are provided by third parties (e.g. the infrastructure, middle agents, etc.):

- *Service Matchmaking (or Discovery)*: a service (provider) is located for a given service request.
- *Service Composition Planning*: if no provider is found then a composite service is built from existing services.

In order to support these two functionalities, services are described and registered in a directory (yellow pages).

In the following we present our approach to enriching web service descriptions with organisational information. For this purpose, we first introduce simple languages for representing role-based service advertisements and service requests.

Definition 2. A service advertisement S is a set of pairs so that

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

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).

For a second opinion service, the service advertisement could be *{<ar, ir>, <er, ->, <ir, ->}*. For instance, in order to play the advisor role (*ar*), it is necessary that the requester may take on an informer role (*ir*) so as to be able to provide additional information.

Definition 3. A service request Q is a pair so that

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

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).

An example of a second opinion service request could be *<ar∧er, {ir, er}>*, where the requester is looking for a service provider capable of engaging in advisement and explanation interactions, while announcing that it may provide information and explanation capabilities.

Although organisational information is not a first-class citizen in service description languages such as OWL-S[1] or WSMO[2], it is not difficult to incorporate it into them. In OWL-S, for instance, we propose to include the role description as an additional parameter, called *Service_Roles*, in the case of service descriptions (r and ρ are mapped to the *providerRole* and *dependingRoles* tags, respectively), and *Query_Roles* for service requests (ρ and C are mapped to *SearchedProviderRoles* and

[1] http://www.daml.org/services/owl-s/

[2] http://www.wsmo.org/

CapabilityRoles). Figure 2 shows an excerpt of a service description in OWL-S.

```
<profile:ServiceParameter rdf:ID="SERVICE_ROLES">
 <profile:sParameter>
  <roles:ServiceRoles rdf:ID="ROLE_DESCRIPTION_LIST">
   <roles:interactiveRole>
    <roles:InteractiveRoleDescription rdf:ID="INTERACTIVE_ROLE_1">
     <roles:providerRole rdf:resource="http://.../roles.owl#AdvisorRole"/>
     <roles:dependingRoles rdf:resource="#DEPENDING_ROLES_1"/>
    </roles:InteractiveRoleDescription>
   </roles:interactiveRole>
   <roles:interactiveRole>
    <roles:InteractiveRoleDescription rdf:ID="INTERACTIVE_ROLE_2">
     <roles:providerRole rdf:resource="http://.../roles.owl#ExplainerRole"/>
    </roles:InteractiveRoleDescription>
   </roles:interactiveRole>
   <roles:interactiveRole>
    <roles:InteractiveRoleDescription rdf:ID="INTERACTIVE_ROLE_3">
     <roles:providerRole rdf:resource="http://.../roles.owl#InformerRole"/>
    </roles:InteractiveRoleDescription>
   </roles:interactiveRole>
  </roles:ServiceRoles>
 </profile:sParameter>
</profile:ServiceParameter>

<roles:RoleExpression rdf:ID="DEPENDING_ROLES_1">
 <roles:item>
  <roles:ConjunctiveRoleList rdf:ID="CONJUNCTIVE_ROLE_LIST_1_2">
   <roles:roleEntry rdf:resource="http://.../roles.owl#InformerRole"/>
  </roles:ConjunctiveRoleList>
 </roles:item>
</roles:RoleExpression>
```

Figure 2 – Second opinion OWL-S service profile (partial)

3. SERVICE MATCHMAKING

In this section we describe a matchmaker based on organisational information, which can be used as a complement to standard I/O based matchmakers. We first describe our role-based matching algorithm [5] that takes as inputs a service request (R) and a service advertisement (S) and returns the degree of match (*dom*), and sketch its implementation. We then perform an analysis of the performance of our role based matchmaker.

3.1 Role-based Matchmaker

Motivated by current trends in notions of match and concept similarity techniques, we set out from the following requirements to define a semantic match function between two roles:

1. It must return a real number in the range [0..1], with a higher value the more similar the concepts are (1 if $r_1=r_2$).
2. It must consider the distance between both concepts (roles) in the ontology: the greater the distance, the less similar are the concepts (decreasing function).
3. The change of *dom* per unit must decrease inversely with the distance (e.g., the step from 1 to 2 is more relevant than 5 to 6).
4. The $dom(r_1,r_2)$ must be independent of the height of the taxonomy and its location within it.
5. The logical relation between the two roles (i.e. the subsumption relation) must be taken care of. This is the most important criterion to take into account.

Requirement 2 is addressed by using the measure proposed by Rada [14], consisting of the number of edges in the shortest path between two concepts in the taxonomy:

$$dist(c_1,c_2) = depth(c_1) + depth(c_2) - 2\times depth(lcs^3(c_1,c_2))$$

Requirement 3 imposes a non-linear decreasing function. We use a typical exponential function here, $e^{-dist(r_1,r_2)}$, as it maintains its range in [0..1], is monotonically decreasing, is 1 when $r_1=r_2$ (requirement 1), and it does not depends on the height of the taxonomy nor the global height of them (requirement 4).

In order to fit requirement 5, we differentiate among the four levels of match proposed by Paolucci et al. [13] (advertisement *A* and request *Q*):

- *exact*: if $r_A = r_Q$
- *plug-in:* if r_A *subsumes* r_Q
- *subsumes*: if r_Q *subsumes* r_A
- *fail*: otherwise

We must combine the numerical value (req. 1-4) with the level of match, which has higher priority. We take the final value, representing the degree of match, 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*. So we only have to scale the value [0..1] to the ranges [0..0.5] and [0.5..1]. These considerations lead to the following equation:

Definition 4. The degree of matching *dom* between two roles R_A and R_Q is given by

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

where $\|R_A,R_Q\|$ is the distance between R_A and R_Q ($dist(R_A,R_Q)$) in the ontology *O* (if there is a subsumption relation between them). By construction, this equation fits the requirements.

The *semantic match between two services* is done by searching the role in the advertisement *S* that best matches the one in the query (*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]).

Figure 4 shows the algorithm that we have developed to determine the degree of match (*dom*) between a service request

[3] Least common subsumer

(*Q*) and a service advertisement (*S*), and that constitutes the nucleus of our role-based matchmaker called ROWLS. Our implementation relies on the Mindswap Java Library[4] for parsing OWL-S service descriptions, and on Jena[5] for managing OWL ontologies.

```
Match(Q: service request, S: service advertisement)
    dom = 0
    FOR ALL CRSi IN Q.ρ
        dom' = ∝
        FOR ALL rj IN CRSi
            dom' = min(dom',MatchAtomicRequest(rj,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(role,r)
        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' = ∝
        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(role,ri))
    return dom
```

Figure 4 – Role-based matching algorithm

3.2 Evaluation

We have realized several experiments to evaluate the efficiency (response time) and the effectiveness of our approach. For our testing, we used a subset of the OWLS-TC v2[6]. We annotated selected service descriptions with organisational information (roles) and added new services and queries, leading to a set of 378 OWL-S services that were used in our experiments.

As mentioned above, the approach presented in this paper is intended to be complementary to other general-purpose matchmakers. In our experiments, we combined ROWLS with OWLS-MX [9], one of the leading hybrid matchmakers available to-date. The results reported in the next subsections essentially compare relevant features of a combination of ROWLS and OWL-MX compared to a standalone use of the latter.

[4] www.mindswap.org/ mhgrove/kowari

[5] http://jena.sourceforge.net

[6] http://projects.semwebcentral.org/projects/owls-tc/

3.2.1 Efficiency Evaluation

Figure 3 depicts the results of the scalability tests that we have performed to evaluate ROWLS[7]. It shows that the matching time increases linearly with the number of services, giving an average of 0.03 ms in matching a query with one service. This is two

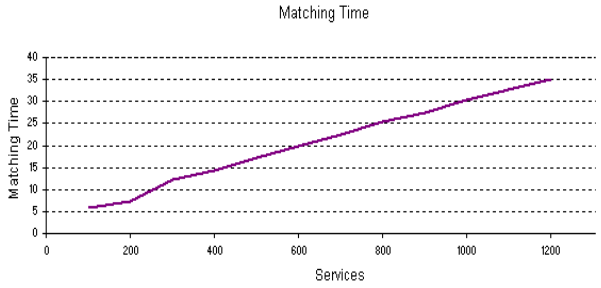

Figure 3 – Matching time (ms)

orders of magnitude less than OWLS-MX matching time. We explain this by the fact that ROWLS concentrates on a smaller number of characteristics of a service description. In addition, the ontology used by ROWLS is smaller than the domain ontologies that OWLS-MX has to deal with. Load time was not considered in the scalability test because it depends on several external factors like disk access time, network connections, web servers, etc.

3.2.2 Effectiveness Evaluation

Another relevant question is as to how far the use of ROWLS can improve the results of a general-purpose matchmaker in terms of effectiveness. In particular, we were interested in measuring the performance of an architecture in which ROWLS is used as a filter which eliminates a number of services, so that they need not be fed as input to OWLS-MX. For this purpose, our tests were carried out with different ROWLS filter configurations that pass on only a certain percentage of the best-ranked services (according to role-based matching) to OWLS-MX (30%, 60%, 90% and 100%, respectively).

Our first experiments shed light on the different relations between *precision* and *recall* (measures widely used in the valuation of information retrieval systems). Figure 8 shows the (macro-averaged) precision-recall-curves for OWLS-MX alone and for the four combinations of ROWLS and OWLS-MX. As shown in the figure, based on our test collection, using ROWLS as a filter for the general-purpose OWLS-MX matchmaker yields to better precision compared to a standalone version of OWLS-MX for all level of recall.

In open multiagent systems with a large number of services, which is the target domain of our research, precision becomes more relevant than recall: we are interested in finding one (or a small set of) service providers, but it is not necessary to retrieve all of them. To account for this fact, we have performed experiments to come up with quantitative data regarding two alternative measures of effectiveness proposed within TREC[8] in such situations, namely *average precision* and *R-precision*.

[7] We used an Intel Pentium 4 3.00GHz computer, with 1GB Ram

[8] http://trec.nist.gov/

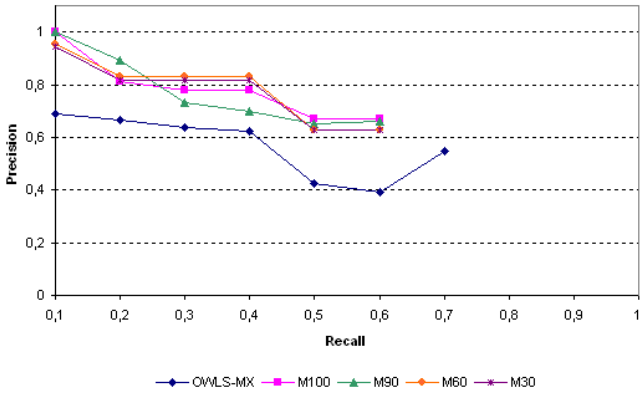

Figure 5 - Precision-recall curves

The *average precision* is shown in Figure 6. It can be seen that the combination of OWLS-MX and ROWLS outperforms a standalone OWLS-MX based on our test collection: the smaller the number of services that pass the filter (ROWLS), the higher the precision. This is because role-based matching of ROWLS is orthogonal to the general-purpose matching strategy of OWLS-MX, thus filtering out irrelevant services that OWLS-MX would have erroneously classified as relevant.

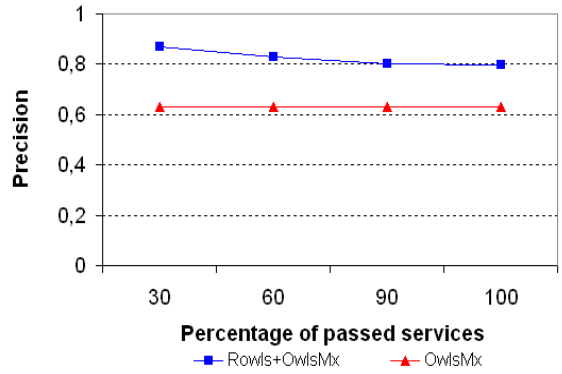

Figure 6 – Average precision

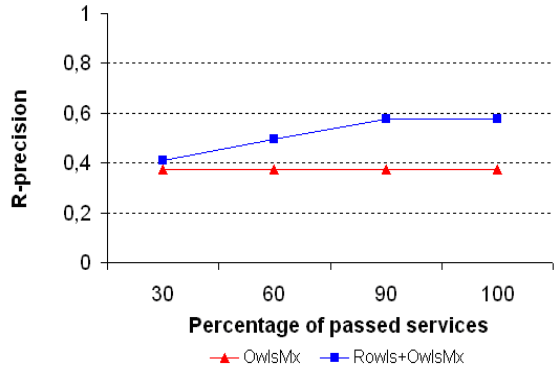

Figure 7 – R-precision

Figure 7 shows the *R-precision* values based on our test collection[10]. Again, ROWLS enhances the behaviour of OWLS-MX. Note that the R-precision increases with the number of services that pass the filter. This happens because only the best ranked services pass the ROWLS filter, and the more services are fed into OWLS-MX, the higher the probability that irrelevant services are among them.

4. FILTERS FOR COMPOSITION

In SOMAS middle agents provide different kinds of matchmaking functionalities. If no adequate services are available for a specific request, a planning functionality can be used to build up composite services. This problem has subtle differences with the classical AI planning problems as service composition plans need not be very deep but, in turn, can be built up from a vast number of services (operators) that are usually registered in the directory. In order to take advantage of recent advances in the field of AI planning for this purpose, we propose exploiting the organisational information available in SOMAS to heuristically filter out those services that are probably irrelevant to the planning process. In this section, we first present an abstract framework for service-class based filtering. We then show how it can be instantiated to a particular MAS domain based on role and interaction ontologies, and finally present a quantitative evaluation of this approach.

4.1 Generic Filtering Framework

At a high level of abstraction, the service composition planning problem can be conceived as follows: let $P = \{p_1, p_2, \ldots, p_m\}$ be the set of all possible plans (composite services) for a given service request R, and $D = \{s_1, s_2, \ldots, s_n\}$ the set of input services for the proper service composition planner (i.e. the directory available). The objective of a filter F is to increase the computational efficiency of the planning process by reducing the search space. Therefore, it selects a given number $l<n$ of services from D, such that:

- the number of plans from P that are excluded from the search space is minimised and, in any case,
- at least one plan of P can still be found.

Our filtering framework makes use of *service class information* so as to cluster services based on certain properties. We use the following filtering heuristics to determine the relevance of a service of class s for a certain service request of class r:

1. the higher the *frequency* of s in past plans that were generated for requests of class r, the higher the relevance of services belonging to that class s;
2. the smaller the *dimension* of past plans that were generated for requests of class r and that s was necessary for, the higher the relevance of services belonging to that class s;

[10] Precision after retrieving R services, being R the number of relevant services of the test collection

Definition 5. Suppose a service description language L_{SD} and a query language L_Q, a filter F over L_{SD} and L_Q is a triple $<C, A, R>$, where:

- C is a set of all *classes of services*
- A: $L_{SD} \rightarrow 2^C$, is a function that for each service description $s \in L_{SD}$ returns the classes $A(s)$ that it belongs to.
- R: $L_Q \rightarrow \Phi$, is a function that for each service request $r \in L_Q$ returns a formula $\phi \in \Phi$ describing the requested classes, of the form

$$\phi = \bigvee_{i=1}^{n} \bigwedge_{j=1}^{m} c_{ij}, c_{ij} \in C$$

Figure 9 depicts the structure of our approach to service composition filtering. With each outcome of a service composition request, a Historical Information Matrix H is updated. Setting out from this information, a Relevance Matrix v is revised and refined. Based on this matrix, service relevance can be determined in a straightforward manner. For each service composition request, a filtering method is invoked which, in turn, is based on this notion of service relevance.

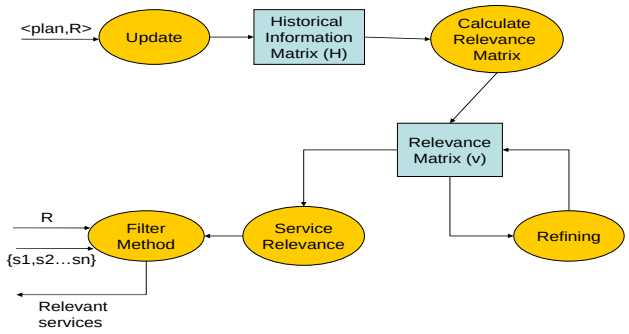

Figure 9 – Architecture of the filter component

The *Historical Information Matrix* for a service class r compiles relevant characteristics of plans (composite services) that were created in the past in response to requests for services belonging to that class. In particular, for each plan dimension i and service class C it records the number of plans of length i that made use of services of class C. Historical Information Matrixes are updated as newly generated plans come in. If the service request is a logical formulae (given in disjunctive normal form), we *distribute* the contribution of the resulting plan among the affected Historical Information Matrixes [6].

The Relevance Matrix specifies the relevance of a service class s to be part of a plan (composite service) that matches the query for a certain service class r.

Definition 6. Let s be a service class and r a class included in a service request, the *relevance* of s with respect to r ($v(s,r)$) is a value between 0 and 1 calculated from the information about plans (*Historical Information Matrixes*) as:

$$v(s,r) = \sum_{d=1}^{m} \frac{n_d}{d^\alpha} \Big/ \sum_{d=1}^{m} \frac{N_d}{d^\alpha}$$

Where d is the dimension of the plan, m is the dimension of the longest plan stored, n_d is the number of times that s was part of a composite plan of dimension d for the request r, and N_d is the total number of plans of dimension d for that request. α is a constant > 0 that allows giving more importance to plans of smaller dimension.

The Relevance Matrix $v(s,r)$ can be further refined in order to take transitivity into account [6].

Definition 7. Let $v(s,r)$ be a relevance matrix, its *k-step refined matrix*, $v^k(s,r)$, is calculated as:

$$v^1(s,r) = v(s,r)$$

$$v^k(s,r) = Max\ (v^{k-1}(s,r),\ v^{k-1}(s,s_1) * v^{k-1}(s_1,r),\ v^{k-1}(s,s_2) * v^{k-1}(s_2,r),\ \ldots,\ v^{k-1}(s,s_n) * v^{k-1}(s_n,r))$$

The first step to calculate the *relevance* of a service s for a request r is the mapping of both to *classes of services*. Then, the relevance between the classes is calculated. We will use $v(s,r)$ to represent the relevance of *class s* for the *class r* in the request, and $V(S,R)$ as the *relevance* of service S for the service request R.

Considering that, in general, a service S may belong to *several* classes (s_1, s_2 … s_n), if a request R only includes a class (r) in its description: $V(S,R) = \max(v(s_1,r), v(s_2,r), \ldots, v(s_n,r))$.

However, if the request specifies a logical expression containing several classes of services (r_1, r_2 … r_m), we evaluate logical formulas using the *maximum* for disjunctions and the *minimum* for conjunctions; and inside the *maximum* is used to aggregate the service classes specified by the provider. For example, if the request R includes the formula $r_1 \vee (r_2 \wedge r_3)$, and the service S belongs to the classes s_1 *and* s_2, the calculus is as follows:

$$V(S,R) = \max[\ \max(v(s_1,r_1), v(s_2,r_1)),\ \min(\max(v(s_1,r_2), v(s_2,r_2)), \max(v(s_1,r_3), v(s_2,r_3)))\ ].$$

We have build three different types of filters based on the above approach, depending on whether they return only services whose relevance exceeds a certain *threshold*, whether they are among the *k best* services, or whether they are within a certain *percentage* of the most relevant services. In addition, every such filter allows specifying a certain probability by which services are selected randomly to assure an adequate exploration of the plan space [6].

4.2 Role-Based Filtering

Our role-based filtering method relies on the organisational information specified in Definition 1. The idea is to relate roles searched in the query to roles played by agents in the composite service, that is, what are the roles typically involved in a plan when a role r is included in the query. For example, it is common that a *medical assistance* service includes *travel arrangement, arrival notification, hospital log-in, medical information exchange* and *second opinion* interactions.

A filter instance is specified by defining the three components of Definition 5. In particular, in the case of the role-based filter we use the information described in section 2.2 (Definition 2 and Definition 3).

Definition 8. A Role-based filter is a triple F = <C, A, R> where:

- *C* is the set of roles in the ontology ($O = <R, I, \leq, \Delta>$), i.e., $C=R$.
- *A* is the function that, given a service description *S* returns the set of roles its provider can play, i.e. $A(S)=\{r|<r,\rho>\in S\}$. This information is extracted from the *Service_Roles* OWL-S service profile parameter.
- *R* is the function that, given a service request *Q*, returns the role expression that defines the query, i.e. $R(Q)=\rho$ (recall that service request $Q=<\rho,C>$).

4.3 Evaluation

Let *S* be a set of possible services in a SOMAS. Let P_S^q be the set of all plans that can be composed for the query *q*, and $P_S=\cup P_S^q$ the set of all possible plans with services from *S*. However, in a particular instant we consider a directory *D*, which contains a subset of the total set of services $S_D \subseteq S$, being P_D the set of plans from S_D. After applying the filter *F*, only the set $S_D^F \subseteq S_D$ is passed on to the planner. The goal of our tests is to evaluate how the filter affects the plan space.

Despite several initiatives, a suitable test collection for semantic web service composition planning is still to come. In our case, we also need to include role-based annotations. For this reasons, we created a test collection as follows: We set out from a graph of services, whose vertices represent world states, and whose edges represent services (set *S*) and the corresponding state change. A plan for a given service request *q* can be seen as a path from one node to the other of the edge that represents the service. Following this approach we generate the set P_S of all possible plans by calculating all the paths between the two nodes connected by every edge. This set is used to train the filter.

Figure 10 shows several curves, each of them corresponds to a percentage of services in the directory (i.e. $|S_D|/|S|$) and represents the ratio of maintained plans for different percentages of filter. As expected, the more services pass the filter, the more plans can be composed. It is interesting to observe that the smaller the directory, the better results obtained, except for very high filter percentage. Note that the less services are considered, the less plans can be composed with them. In this situation the plan heuristic that gives preference to shorter plans, as well as the refinement steps for the relevance matrix become more important. In particular, notice the excellent behaviour when there are less than 70% of the services in the directory.

In addition to the completeness ratio, it is also important to assure that, despite filtering a number of services, it is still possible to find at least one plan. This can be relevant as in some situations (e.g. limited resources) it might be important reducing the completeness ratio but that one plan can be found. shows this analysis, with a ratio close to 1 until a very high percentage filter is applied.

Besides those two main analysis, we have also checked that the filter performs better by refining the relevance and considering the heuristic of plan length (in particular α=3).

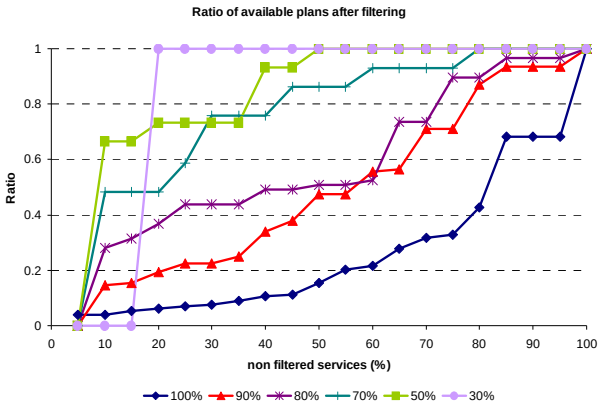

Figure 10 – Ratio of available plans after filtering

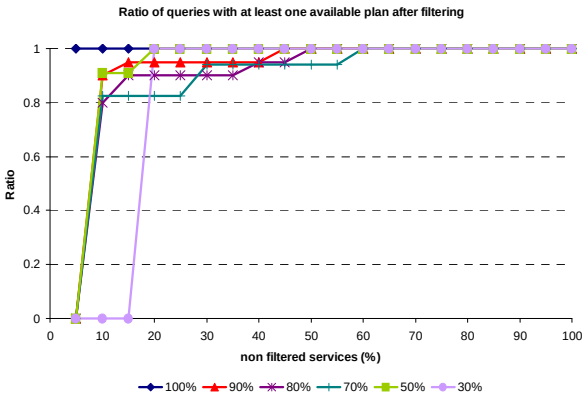

Figure 11 – Ratio of queries with at least one available plan after filtering

5. CONCLUSIONS

In this paper, we have argued that organisational information should play an important role in service-oriented MAS. To back that claim, we have first proposed a method to annotate service descriptions with information about the roles that service providers play when engaging in different types of interactions, and we have shown how this information can be represented in OWL-S. Setting out from this basis, we have provided evidence for the value of this kind of information for service coordination along two major lines.

Firstly, we have developed a novel role-based matchmaking mechanism that draws heavily upon organisational information contained in service descriptions. This mechanism is orthogonal to general-purpose service matching techniques that focus mainly on inputs and outputs of services ([9],[11],[13]). We have implemented a role-based matchmaking component and evaluated it in combination with the well known OWLS-MX matchmaker [9]. Our experiments have shown that the combination of both matchmakers outperforms a standalone version of the OWLS-MX matchmaker in both efficiency and effectiveness.

Secondly, we have proposed a generic filtering framework for service composition, and have implemented a role-based filter component. We have carried out experiments that demonstrated an excellent behavior in terms of completeness (ratio of available plans) and ratio of queries with at least one possible plan.

Though limited in number, there are some other filtering approaches to composition planning in recent literature. However, to the best of our knowledge, none of them makes use of organisational information within a service-oriented MAS to filter services. In [16] an interactive filter that helps the user to compose a plan is described. The system provides the user with the list of services that share an input or output with the current uncompleted plan. Constantinescu [2] proposes an approach to integrate in the directory selection and ranking functions that can be exploited by a planning algorithm. Synthy [1] includes a filtering method similar to ours in the sense that it is applied just before feeding the planner, but they consider relevant those services that can contribute to the goals (at least an effect is shared with a goal), or the preconditions of any service that potentially could contribute to the goal.

Both the matchmaker and filter components have been incorporated into the service-oriented multiagent platform developed by the EU project CASCOM[11]. In particular, the role-based matchmaker has been integrated with OWLS-MX [9] into a Service Matchmaker Agent (SMA), while the role-based filter component is part of the Service Composition Planning Agent (SCPA) together with OWLS-XPlan [10], a heuristic hybrid search AI planner for the composition of OWL-S services. Both agents are part of a software demonstrator in the field of medical emergency management, which has shown excellent performance in several real-world trials.

In the future we plan extend the type of organisational information to be used by service coordination mechanisms. We also expect to confirm the excellent behaviour of our role-based matchmaker when used in conjunction with other matchmakers different from OWLS-MX. Finally, we intend to perform an experimental comparison of our role-based method to other service composition filter instantiations, in particular, those based on OWL-S Service Category information.

6. ACKNOWLEDGMENTS

This work has been partially supported in part by the European Commission under grant FP6-IST-511632 (CASCOM), and by the Spanish Ministry of Education and Science through projects "Agreement Technologies" (CONSOLIDER CSD2007-0022, INGENIO 2010), TIN2006-14360-C03-02 and URJC-CM-2006-CET-0300.

7. REFERENCES

[1] V. Agarwal, G. Chafle, K. Dasgupta, N. M. Karnik, A. Kumar, S. Mittal, and B. Srivastava. Synthy: A system for end to end composition of web services. *J. Web Sem.*, 3(4):311–339, 2005.

[2] W. Binder, I. Constantinescu, and B. Faltings. Directory support for large-scale, automated service composition. In *Software Composition*, volume 3628 of LNCS, pages 57–66. Springer, 2005.

[3] Cáceres, C., Fernández, A., Ossowski and S., Vasirani, M. Agent-Based Semantic Service Discovery for Healthcare: An Organizational Approach, *IEEE Intelligent Systems*, vol. 21, no. 6, pp. 11-20, Nov/Dec, 2006

[4] Cavedon, L., Maamar, Z., Martin, D., Benatallah, B.: Extending Web Services Technologies: The Use of Multi-Agent Approaches. Springer (2004)

[5] Fernandez, A, Vasirani, M., Caceres, C., and Ossowski, S.: A Role-Based Support Mechanism for Service Description and Discovery, in *Service-Oriented Computing: Agents, Semantics, and Engineering*, J. Huang et al. Eds. LNCS 4504 Springer-Verlag, 2007, pp. 132-146.

[6] Fernandez, A, and Ossowski, S.: Semantic Service Composition in Service-oriented Multiagent Systems: A Filtering Approach, in *Service-Oriented Computing: Agents, Semantics, and Engineering*, J. Huang et al. Eds. LNCS 4504 Springer-Verlag, 2007.

[7] Huhns, M. N., Singh, M. P.: Service-Oriented Computing. John Wiley & Sons (2005)

[8] Huhns, M. N., et al: Research Directions for Service-Oriented Multiagent Systems. *IEEE Internet Computing*, 9 (6), 2005.

[9] Klusch, M.; Fries, B.; Sycara, K.: Automated Semantic Web Service Discovery with OWLS-MX. *Proceedings of 5th International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*. Hakodate, Japan. ACM Press. 2006.

[10] Klusch, M., Gerber, A., and Schmidt, M. Semantic Web Service Composition Planning with OWLS-XPlan. *Proceedings 1st Intl. AAAI Fall Symposium on Agents and the Semantic Web* (Arlington VA, USA, 2005).

[11] Li, L., and Horrock, I.: A software framework for matchmaking based on semantic web technology. In *Proc. 12th Int World Wide Web Conference Workshop on E-Services and the Semantic Web (ESSW)* (2003).

[12] Luck, M., McBurney, P., Shehory, O., S. Willmott: Agent Technology: Computing as Interaction (A Roadmap for Agent Based Computing), AgentLink, (2005).

[13] Paolucci, M., Kawamura, T., Payne, T., and Sycara, K.: Semantic matching of web services capabilities. In Proceedings of the *First International Semantic Web Conference on The Semantic Web*, Springer-Verlag (2002) 333-347

[14] R. Rada, H. Mili, E. Bicknell, and M. Blettner. Development and application of a metric on semantic nets. IEEE Transactions on Systems, Man and Cybernetics, 19(1):17–30, 1989.

[15] Serrano, J.M.; Ossowski, S.: A compositional framework for the specification of interaction protocols in multiagent organizations. *Web Intelligence and Agent Systems*, 5 (2). IOS Press. 2007.

[16] E. Sirin, B. Parsia, and J. A. Hendler. Filtering and selecting semantic web services with interactive composition techniques. *IEEE Intelligent Systems*, 19(4):42–49, 2004.

[17] Zambonelli, F., Jennings, N. R., and Wooldridge, M. Organizational Abstractions for the Analysis and Design of Multi-agent Systems. *Agent-Oriented Software Engineering: First International Workshop, AOSE 2000*, (Limerick, Ireland, June 10, 2000). 235-251

[11] http://www.ist-cascom.org