An Agent-based Matchmaker

(*A case study in biomedical services discovery*)

Flavio Corradini, Chiara Ercoli, Emanuela Merelli and Barbara Re Dipartimento di Matematica e Informatica,
Università di Camerino
62032 Camerino, Italy
Email: {flavio.corradini,emanuela.merelli}@unicam.it, {chiara.ercoli,barbara.re}@studenti.unicam.it

***Abstract*— Service discovery is the process of localizing resources and services available in large scale open and distributed systems. In a distributed and redundant system as the Web, it is necessary, beside localizing services, to filter them in order to obtain those which are best for the activities for which they have been requested. By the term *matchmaker* we mean a software entity which monitors services availability, maintains an updated file of all useful information for using services and possibly ensures a quality choice of them. In this paper we propose an architecture for an agent-based *matchmaker*. The matchmaker that takes part in the request process has been developed by using the potential of a *quality model* based on suitable parameters to ensure the proper choice of a service to be consumed in a specific application domain. A case study in biomedical domain is presented. This case study is concerned with the development of a multi-agent system including a Bio-certifier in support of service discovery activity.**

I. Introduction

Service discovery is the process of localizing services and resources in the Web that best fit the requests of potential users.

The Web main feature is the interconnection of an ever increasing number of open, dynamic and geographically distributed systems which have an high heterogeneity of resources, information systems and tools for specific application domains. Hence the Web is a rather complex environment for service discovery activities as can be seen, for example, in the biomedical domain.

Biological and medical research is characterized by a global distribution of information and by an almost complete autonomy of research groups, from which an heterogeneous, redundant, incomplete and rapidly aging access to resources derives. Hence the choice of what could be the most suitable tool or service for biomedical work activities is often difficult and time consuming. From these considerations there follows the need of building a quality model to support the discovery process which will be based in symbolic descriptions of relations among concepts of one or more domains of interest allowing classification of services which are functionally similar [1].

Quality can be defined as all the features of an entity (resource, service, tool) that influence its capability to satisfy declared or implicit needs [2]. From this definition it is clear that it is difficult if not impossible, until today, to define a specific metrics capable of measuring the quality of resources available through the Web. Although there exists several criteria to evaluate consistency and internal correctness of a resource, true evaluation of its quality, that of interest to users, relies on the effectiveness of the resource itself. In other words, one has to ascertain if a specific group of users considers the use of that resource satisfactory for its information needs. In fact, before finding the ideal requirements for the quality model, it is necessary to carefully analyze the application domain in which the quality model has to be used. Hence the quality model has two main components, the general one which describes the quality aspects of the distributed system, e.g. the Web, and the other which describes the specific quality aspects of the application domain; the biomedical in our case study. The established quality model then becomes a tool of consultation for the software entity in charge of service discovery.

In this paper an architecture for a quality of service (QoS) agent-based matchmaker is presented. The term *matchmaker* [3], [4] means a software entity capable of monitoring the availability of services, maintaining an updated file of all information on service use and, we add, of providing a quality choice of service. The matchmaker is an agent contacted by other agents wanting to obtain a quality service with respect to the activity where the service will be used. In order to ensure a choice of quality of a requested service, the matchmaker communicates with the QoS *certification authority*, i.e. an agent capable of implementing the established quality model.

Briefly, an agent is a software system capable of acting with the aim of solving a problem in a flexible and autonomous way and in an open, dynamic, unpredictable environment which is typically populated by other agents. Often agents are created to interact and cooperate with each other. The need of making an agent interacting and communicating with other agents leads to the need of coordinating the activities of the agents involved in a system [5], [6]. In order to coordinate a pool of agents (MAS: Multi-Agent System) it is necessary and fundamental to understand which are the actors involved in the system, their roles and which information are more important. In so doing, we also achieve the result of specifying the importance and true value of the parameters that characterize the quality model.

The coordination model we have followed is the matchmaking model presented in [7] which is based on a process of mediation that implements direct communication among the providers and the consumers of services and resources.

This work was supported by the by the MURST strategic project 'Oncology Over Internet', by the CIPE project 'SICOM: Sistemi Coperativi Multi-agente' and the Center of Excellence for Research 'DEWS: Architectures and Design Methodologies for Embedded Controllers, Wireless Interconnect and System-on-chip'.

M. Baldoni, F. De Paoli, A. Martelli, and A. Omicini (Eds.): *WOA 2004 – Dagli Oggetti agli Agenti, Sistemi Complessi e Agenti Razionali*, Torino, Italy, November 29th – December 1st 2004. Pitagora Editrice Bologna. ISBN 88-371-1533-4. http://woa04.unito.it

To the aim of showing the applicability of matchmaker architecture enriched with the QoS component, we have examined a case study in the biomedical domain, and developed a multi-agent system for the discovery of quality services based on JADE[1] platform.

The remaining of the paper is organized as follows. Section II is an introduction to the case study. Section III presents the architecture of the multi-agent system for service discovery and introduces the quality model. Section IV describes the architecture of the system, defines the quality model for biomedical domain and debates some experimental results. In section V different approaches proposed in the literature for the service discovery are analyzed and future extension of the paper are presented.

II. A Case Study in the Biomedical Domain

Health science is the applied science discipline that deals with human and animal health by means of study, research and application of knowledge with the aim of improving general health. Biomedicine is a branch of health science that applies biological and physiological principles to clinical practice. Support to biomedicine is given by the understanding of the way in which both human and animal biological systems work and by the analysis of the (sometimes hidden) existing relations between medical reports and results of performed therapies. In both cases, the use of computational tools allows us to find and analyse biological and clinical information in order to answer complex questions in biological domain. Moreover, appropriate computational models would also allow to simulate biological systems [8], [9] with the aim of verifying properties useful both for diagnosis and therapy.

The Web is an endless source of information of fundamental importance to increase knowledge in the biomedical domain, however it is often difficult and complex to retrieve this information.

In several disciplines, complex questions are stated by means of workflow of activities representing different instances of problems which are simpler to solve. Carrying out these activities implies the use of resources (tools and services) usually accessible through the Web [10]. Introducing quality in a workflow means giving a way of finding the most appropriate resource/service to effectively satisfy the requests of each activity. The following two scenarios are presented as examples in order to introduce the workflow concept in biomedicine:

"Let us assume that a biomedical researcher be an expert of a gene, of the corresponding protein, of the known mutations of that protein, and of consequent pathologies as well. This biomedical researcher wants to design a microarray[2] experiment to analyse the gene expression (i.e. how much the gene produces) in different normal and pathological tissues. This experiment allows him to also find out the genic expression of other genes in addition to the one of the gene being studied and hence he needs to have an updated list of the genes that could be involved in the same biochemical pathway (i.e. chain of biochemical reactions). For this purpose, the biomedical researcher decides to use the Gene Ontology (GO) annotation[3] to find out the relations among genes, biological processes and biochemical pathway." In this example GO is used as a domain-specific language to specify the request, thus GO terms will effect the domain specific quality aspects of the proposed model.

"A doctor is treating a patient that has some constant slight temperature (37,5 C). The temperature persists after antibiotics therapy and hence the doctor decides to control the protein level of the patient and prescribes some blood tests and urine test. The performed tests show that the level of some proteins is not normal but no sure conclusion can be drawn (no certain diagnosis). The doctor then decides to search the possible interactions among these proteins." Instead, this example does not use any domain specific language (ontology) to describe the service, thus the quality model consists only of the general quality aspects.

In these and similar situations the search for useful information with the aim of giving an answer to the questions being asked implies the choice and use of several resources. The discovery process should be capable of identifying the best service which will give the sought result in the shortest time, so making the system efficient and effective.

III. The Multi-Agent Architecture for Service Discovery

In this section we present the architecture of the multi-agent system and the quality model defined to support the service discovery in a distributed environment.

The system supporting service discovery has been designed using agent technology because the problem dealt with was suitable to be described in terms of autonomous, flexible actors which operate in a dynamic and unpredictable environment and are created to cooperate with each other. Our choice has also been affected by such parameters as development and administration costs of the discovery system, implementation of interoperability among the different active systems, and guarantee of an acceptable security level.

The proposed system architecture is an extension of the one defined in Retsina [11] infrastructure and its main feature is a group of three actors (agents) that communicate and exchange information among themselves with service discovery as their common goal.

The *service provider* supplies the services by which it is possible to find the required information or solve specific computational problems related to an application domain. The *service requester* is the user (or consumer) of the services offered by the *service provider*, and finally the *middle-agent* is the software entity that mediates between the previous two in order to find the sought services.

In the literature [11] *middle-agents* are classified according to their functionalities as mediators, brokers and matchmakers. In its original definition, given by Wiederhold in 1992 [12], a *mediator* is the active and dynamic interface by which a given

[1] http://jade.tilab.com

[2] A DNA microarray is a piece of slide with a microscopic array on which single DNA pieces are placed.

[3] www.geneontology.org/

user (*service requester*) access data and knowledge owned by itself.

A *broker*, sometimes also called a *facilitator*, is the existing interface between a *service requester* and a *service provider* which acts as a mediator for service requests. All communications between pairs of *service providers* and *service requesters* flow into the broker which typically contacts the more important *service providers* negotiating execution of and access control to the most appropriate service and returning the service result to the *service requester*.

On the contrary, the task of a *matchmaker* is to create a connection between the *service requester* and the *service provider*, matching the request of a given service requester to the offer of service of a *service provider*. In this case an autonomous interaction will take place. Unlike the functionalities of both the broker and the mediator, the functionality of the matchmaker is to return to the service requesting agent an ordered list of *service providers*. Consequently the *service requester* has to directly contact the *service provider* and negotiate with it in order to get the desired service.
One could then consider matchmaking as a subset of brokering but at the same time it can be seen as an extension of it because matchmaking allows the *service requester* a subsequent choice of *service provider* in a way independent of the match found by the matchmaker. The matchmaker has one weak point: each agent needs to be smart enough to form a query and evaluate how to choose among alternative *service providers*, this features being not always present in MAS systems.

The coordination model describing dependencies and interactions between the matchmaker and the other agents is called matchmaking [7]. When an agent publishes a service, the matchmaker records the name of the agent in his knowledge base together with the description of offered service according to the ontology used during the communication act. In this way, when an agent requests a service, the matchmaker looks in his knowledge base for a server capable of satisfying the request (*service matching*). Then the agent requesting service directly interacts with the chosen server in order to get the desired service and data (*service gathering*) so avoiding a possible bottleneck in data transmission or a possible interruption of the matchmaker activity, as described in Figure 1.

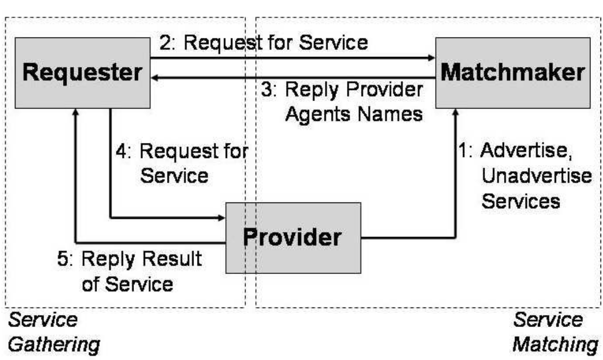

Fig. 1. Matchmaking coordination model [4]

A. The Proposed Agent-based Matchmaker

To the classical matchmaking model, as presented in [11], our architecture introduces a fourth kind of agent (see Figure 2) representing the QoS *certification authority*. This agent, through certification, ensures that resources, services and tools be consistent with the user request non-functional requirements. In the proposed extended model, the duties of the $matchmaker$ also include coordination of available services in accordance with specific protocols, agreements and policies, and mediation to obtain reliable services both in terms of quality and confidence, the latter being due to the multi-agent system.
Such a process of quality service discovery must include a component capable of analysing certain fundamental requirements that are made appropriate to the domain to which they belong. Verification of these requirements will allow the *QoS certification authority* to give a quality level to each registered services taken into consideration. In particular the general evaluation criteria of our authority include some macro-categories as aim of resource, user target, reliability, contents, privacy, updating of formal features and quantitative functions.

The main functionalities of each actor in the dynamics of mediation systems are as follows:

1) a *service provider* advertises its services to a *matchmaker* via WSDL-URI;
2) the *matchmaker* stores this information in an hash table and notifies the new services to the QoS certification authority;
3) the *QoS certification authority* contacts the *service provider* and verifies the quality service;
4) the *QoS certification authority* certifies the service to the *matchmaker* via an XML document;
5) a *service requester* asks *matchmaker* to find a *service provider* that provides the best services;
6) the *matchmaker* processes the request within his knowledge base (collection of information on services and *service providers*) and it yields either some information regarding the *service provider* or possibly the result of the application of the requested service;
7) the *service requester* send the request (service input) to the selected *service provider*;
8) after the executing of service, the *service provider* returns the result (service output) to the *service requester*.

This model, while looking extremely simple at first sight, is instead a rather complex one mainly because the Web is an open system, with plenty of information, subject to continuous changes of available resource location, heterogeneity and contents. The complexity of the model can increase when different user groups and different MAS come into play because each has its own goal which may be in conflict with those of the others.

The choice of integrating matchmaking with a QoS authorization component is a consequence of matchmaking being well suitable to the scenario proposed by the case study, whose features are distributed systems into which agents come in and out and the offering of multiple answers with the possibility

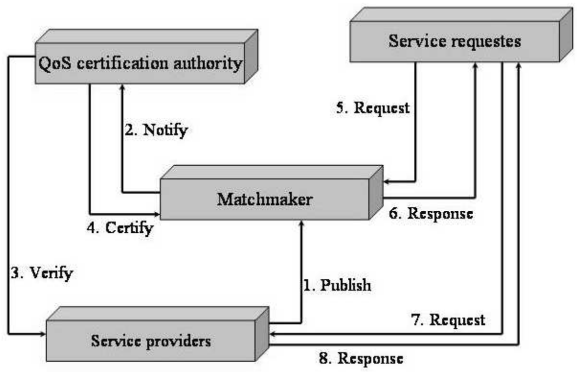

Fig. 2. The MAS architecture extended with the QoS certification authority

for each agent to keep control of its choices. The proposed matchmaker limits the choice among alternative.

B. The Proposed Quality Model

When a user looks for a service (resource, tool, etc), the system ideally should fetch the service exactly matching the one requested. It is practically unlikely that such a service be available and hence a service with “sufficiently similar features” is fetched.

What do we mean by “sufficiently similar”?
In its strongest meaning a service offered in the network and a requested service are “sufficiently similar” when they exactly contain the same functionalities. This is a too restrictive definition since the requesting user does not know in advance how a service is represented in the network and has an own idea of what the service should do. An acceptable definition of similarity can be one with less constraints so to accept a more flexible exactness degree.
Hence localizing services which can be used by the user despite the existing differences between request and offer represents a challenge for the system. Metrics measuring the distance between request and offer can be of help to the user in making a deliberate choice [13].

As above mentioned, the proposed quality model consists of two components, the one describing general quality aspects of the distributed computational environment where the service is offered, and the other including quality features of the application domain. In particular, the quality aspects chosen for the first component have been derived by analyzing the Web, and concluding that a qualitative web resource must provide information to satisfy the following requirements:

- *Aim* is the purpose for which the resource has been developed;
- *User target* is the list of hypothetical users;
- *Reliability* is the probability of successfully using a resource;
- *Feasibility* is the measurement of the easiness to access the resource;
- *Usability* is the measurement of the easiness to use the resource;
- *Originality* is the degree of correctness of the resource and its information;
- *Privacy* captures the legal conditions of using the resource;
- *Updating* is the attendance of the resource updating;
- *Uptiming* is the maximum length of time between two resource failures;
- *Timing* is the daily time of resource activity;
- *Speedy* is the measurement of the execution time;
- *Browsing* is the measurement of the human easiness to find a resource;
- *Popularity* is the number of active consumers;

Each quality aspect above defined is quantitative measured on the basis of several parameters not listed in this work, but available in [14], [15]. While the domain-dependent quality aspects are described in the Section IV-A dedicated to the case study quality model.

Our system draws a distinction among three matching levels:

Exact match is the highest degree of matching and takes place when requests are satisfied by the server with a percentage higher than 90%.

Plug-in match takes place when a service more general than the requested one is supplied but that can be used instead of the ideal requested service. This kind of matching happens when requests are satisfied with a percentage between 10 and 90%.

Relaxed match is the lowest degree of matching and takes place when requests are satisfied by the server with a percentage lower than 10%.

The matching algorithm measures the distance between the quality aspects and the user requirements for a request service. The matching algorithm developed in this work is carried out within the QoS *certification authority* to support the following actions:

- supporting the semantic matches in a flexible way on the basis of existing ontology;
- achieving matches with a minimum number of positive false matches and negative false ones.
- encouraging correct registrations and requests that take into account the cost of a mismatch due to false declarations;
- carrying out efficient matches that give results in a short time.

The main cycle of the matching algorithm is shown in the code below. It can be seen that the requests are compared with all parameters of the services which are stored in the knowledge base and that the coefficient measuring the degree of matching is evaluated for each service.

```
match (request){
    recordMatch = empty list
    forall service in mirror do{
       recordMatch.addElement(service, coff)
    }
    return best(recordMatch);
}
```

Through our research we have found some general criteria for evaluating quality of resources, services or tools. These criteria can be grouped in macro categories as: purpose, user target, reliability, privacy, updating, formal aspect adherence, interactivity, stability, ease of use and use of and access to established standards.

IV. The QoS Matchmaker in the Case Study

As a case study we have chosen the biomedical domain for its complexity and because from carried out researches come out that quality of Internet medical information is affected by heterogeneity and dispersion of resources and inaccuracy and incompleteness of available information. These factors are well suitable for the definition of a quality model to be integrated in a matchmaker architecture.

Defining a quality model means quantifying the parameters that are typical of the application domain and specifying the framework in which the model will be used. We assume to be in a simplified situation in which biomedical information and services are supplied by information repository which are distributed in the network, called MOBY-Central [16] in our case. Figure 3 shows the components of matchmaker architecture and their interactions.

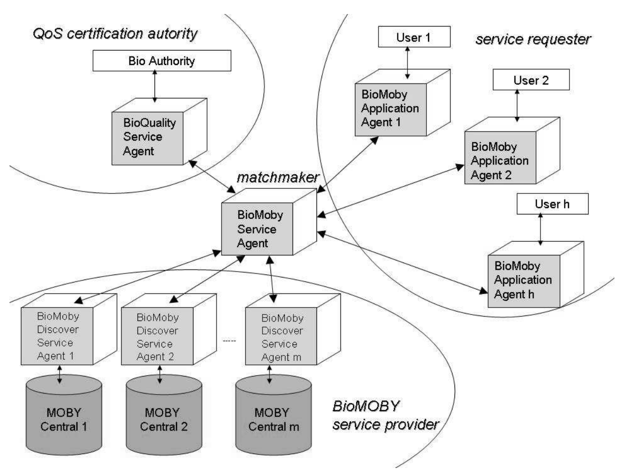

Fig. 3. The matchmaker architecture in the case study

The central element, that is the *matchmaker*, represents the knowledge base of the poll of agents involved in the biomedical system (BioMAS). A *matchmaker* interacts with three separate components (agents): the BioMOBY *service provider* through which increases the system knowledge base which is used in the discovery of new biomedical services, the QoS *certification authority* supporting the discovery service and the *service requester* that carries out the user side application for locating services that satisfy given properties.

In details, the BioMAS consists of four kinds of agents: the `BioMobyServiceAgent`, which is the system main actor because all other agents refer to it. This agent has three different roles: (i) coordinates all other agents; (ii) manages the system knowledge base, and (iii) carries out the discovery of a quality service. The `BioMobyDiscoverServiceAgent` is the interface between the biological information repository and the `BioMobyServiceAgent` with the aims of discovering the requested service on the one side and of advertising any newly offered service on the other side. The `BioMobyApplicationAgent` helps the user in the search of the best service among those advertised by the `BioMobyServiceAgent`. Finally, the `BioQualityServiceAgent` is the authority that certifies services according to the quality model defined for the biological domain.

A. The Quality Model for the Biomedical Domain

The *QoS certification authority* developed in this work used an instance of the quality model introduced in III-B. The first component is characterized by the quality aspects of a Web Services (i.e. `BioMOBY`), while the second is characterized by information introduced by the biomedical domain. The subset of quality aspects chosen for the first component are:

- the *Reliability* based on three parameters: the first one assigns a value to the author based on his professional competence, the second allows to find whether the author adheres to certified standards and the last allows to find out whether the supplier of service is profit oriented;
- the *Originality* based on two boolean parameters: publicity policy, that is whether there are sponsors and official agencies financing the resource and fidelity procedure, that is the monitoring of consumer surveys;
- the *Privacy* based on a boolean parameter that makes sure that privacy policies, data security, personal data processing (including that of unaware users) are in accordance with existing laws;
- the *Updating* based on a parameter that addresses the time period (daily, monthly, yearly) the resource is updated;
- the *Usability* based on a parameter that measures the easiness in using a resource:

Finally, by formal aspect concept we mean two strictly technical parameters which give a measure of the daily service performance:

- the *Timing* that is a measurement of the time period that a service is active;
- the *Speed* that is a measurement of the service execution time.

The matching algorithm, after having analysed the above listed information and after having made a classification of services, goes on to examining the information made available by biomedical domain.

In this second part of the model,

- *name*, represents the most important parameter because the knowledge of it by the user will cause the search necessarily returning the specified service (match weight=51);
- *description*, made of keywords which will be sought inside every individual service stored in the knowledge base (match weight=4);
- *type*, has little importance in the model because can only be one of seven kinds (service, retrieval, resolution, parsing, registration, analysis, NCBI_Blast) (match weight=2);
- *author*, it simply represents his name and does not carry his credentials with it (match weight=4);

- *input* and *output*, they are fundamental parameters because the user already knows what he has got and what he wants to get (match weight=17 and 22);

A more detailed description of the model of quality can be found in [14], [15].

B. Examples

As explained in Section II, as of today the huge amount of information and services in the biomedical domain which are in the Web makes rather difficult to the user to understand which is the best service for his needs. Our contribution to solving this problem has been the implementation of a model based on a qualitative matching algorithm by which it is possible to make the correct choice. Moreover, by the use of an agent based technology, waiting time and interaction time by the user with the system have been considerably reduced because of the presence of a software assistant. In order to show some preliminary results of the effectiveness of the proposed model we will consider the two simple examples previously shown.

In the first case "the biomedical researcher decides to use the Gene Ontology (GO) annotation to find out the relations among genes, biological processes and biochemical courses." The results we would get from BioMOBY by making the following query with keywords 'GO', 'Gene' and 'Ontology':

```
select servicename, url
from service_instance
where description like %Gene%
and description like %Ontology%
and description like %GO%
```

are:

```
servicename: getGoTerm
        url: http://mobycentral.cbr.nrc.ca
             /cgi-bin/Services/Services.cgi

servicename: getSHoundGODBGetParentOf
        url: http://mobycentral.cbr.nrc.ca
             /cgi-bin/Services/Services.cgi

servicename: getSHoundGODBGetChildrenOf
        url: http://mobycentral.cbr.nrc.ca
             /cgi-bin/Services/Services.cgi
```

While the result obtained by the QoS matchmaker is:

```
servicename: getGoTerm
        url: http://mobycentral.cbr.nrc.ca
             /cgi-bin/Services/Services.cgi
```

It can be noticed that in the first case the answer also contains addresses which are not meaningful for the made query forcing the user to a kind of classification or to repeated trials before singling out the service which best fits his needs. In the second case, the QoS machmaker, on the bases of the matching algorithm filters the best service.

Let us analyse the second example "the doctor then decides to search the possible interactions among these proteins." By making the following query to BioMOBY with keywords 'protein' and 'interact':

```
select servicename, url
from service_instance
where description like %interact%
and description like %protein%
```

we would get the following results:

```
servicename: getInteractions
        url: http://www.pdg.cnb.uam.es/moby
             /cgi-bin/mobyservice

servicename: getInteractionsXML
        url: http://www.pdg.cnb.uam.es/moby
             /cgi-bin/mobyservice

servicename: getInteractingMethods
        url: http://www.pdg.cnb.uam.es/moby
             /cgi-bin/mobyservice
```

While, through the middle-agent mediation and use of the model of quality, we obtain:

```
servicename: getInteractions
        url: http://www.pdg.cnb.uam.es/moby
             /cgi-bin/mobyservice
```

Also from this second case, it can be seen that the use of a quality model has the same effect of applying a filter to the set of possible answers.

V. Related and Future Work

Many works have been presented in the literature to support service discovery in the Web environment [3], [4]. Some use UDDI technology and Web Services, others use the agent technology, a few just use a mediator. None of these suggests the integration of a quality model within the matchmaker architecture in support to service discovery in a biomedical domain.

UDDI[4] (Universal Description, Discovery, and Integration) has become a de-facto standard for service discovery in the community of Web Service and it is commonly looked at as a "yellow pages" service. In the UDDI model services are localized through their description by the supplier or by the type of service and both ways of service discovery are built with a limited number of high level sentences that produce a rigid scheme. Although UDDI is a de-facto standard, it does not allow neither a quantitative nor a semantic discovery but only a keyword based search.

Retsina [3] is an open infrastructure for MAS which is capable of supporting communities (oppure populations) of

[4]http://www.uddi.org

heterogeneous agents. Service discovery is based on OWL-S[5] ontological language for service functionality description. The resulting matching process is only a semantic one and not necessarily of quality.

DiscoveryLink[6] is a product developed by IBM that allows a discovery process on many specialized heterogeneous databases by means of a single query that uses specialized wrappers. The resulting system is a rigid one again and forces the user to predefined and limited choices without offering either a semantic service discovery or one of quality.

MyGrid[7] is a pilot project of UK e-Science that provides a middleware open-source Grid developed to supply a virtual workbench in bioinformatics domain. Emphasis is placed on the workflow as an integration tool and on the customisation and source of data. Resources are considered as services that can be statically or dynamically combined within a given framework. However also in this product no quality of the offered service is guaranteed.

We plan to extend this work in the future by customizing requests to target, that is by including in our model of quality parameters which are proper of user profile (computer scientists, biological computer scientists, biologists, etc.). We also mean to add use of ontology in order to describe the user requests both to verify their validity and to correctly describe each service. In fact, GO could also be included in the quality model to describe the bio-domain. In our case study GO represents the service description language.
The introduction and quantification of additional certification parameters will help both the certifying agent and the middle-agent to keep their information updated and hence to answer even complex requests by giving a service workflow. Last but not least, we plan to develop the system in Hermes[8], a mobile agent middleware supporting distributed applications and mobile computing, in order to use mobility to optimize the cost of data transfer and evaluate the possibiltiy to improve the performance of the matchmaker.

ACKNOWLEDGEMENTS

We wish to thank Ing. Lucio Forastieri for the interesting discussions that have led us to the development of this work and Ing. Paolo Romano for having given us useful case studies for validating the proposed model.

REFERENCES

[1] G. Eysenbach and al., "Towards quality management of medical information on the internet: evaluation, labelling, and filtering of information," 1998.

[2] T. C. I. . International organization for standardization, "Iso 8402: Quality management and quality assurance." 1994, vocabulary. 2nd ed. Geneva: International organization for standardization.

[3] K. Sycara, M. Paolucci, M. van Velsen, and J. Giampapa, "The RETSINA MAS infrastructure," *Autonomous Agents and Multi-Agent Systems*, vol. 7, no. 1–2, pp. 29–48, July–Sept. 2003.

[4] K. Decker, M. Williams, and K. Sycara, "Matchmaking and brokering," in *ICMAS-96*, May 1996.

[5] M. Wooldridge and N. R. Jennings, "Agent theories, architectures and languages: A survey," in *Intelligent Agents, ECAI-94 Workshop on Agent Theories, Architectures and Languages*. Springer-Verlag, 1994, pp. 1–39.

[6] N. Jennings and M. Wooldridge, "Application of intelligent agents," in *Agent Technology: Foundations, Applications, and Markets*. Springer-Verlag, 1998.

[7] K. ycara, M. Paolucci, A. Ankolekar, and N. Srinivasan, "Automated discovery, interaction and composition of semantic web services," *Journal of Web Semantics*, vol. 1, no. 1, pp. 27–46, 2003.

[8] H. Kitano, "Systems biology: A brief overview," *Science*, vol. 295, pp. 1662–1664, march 2002.

[9] N. Cannata, F. Corradini, E. Merelli, A. Omicini, and A. Ricci, "An agent-based conceptual framework for biological systems simulation," in *Fourth International Workshop on Network Tools and Applications in Biology*, Camerino, 2004.

[10] F. Corradini, L. Mariani, and E. Merelli, "An agent-based approach to tool integration," *Journal of Software Tools Technology Transfer*, 2004, to appear.

[11] M. Klusch and K. Sycara, "Brokering and matchmaking for coordination of agent societies: A survey," in *Coordination of Internet Agents: Models, Technologies, and Applications*, A. Omicini, F. Zambonelli, M. Klusch, and R. Tolksdorf, Eds. Springer-Verlag, Mar. 2001, ch. 8, pp. 197–224.

[12] G. Wiederhold, "Mediators in the architecture of future information systems," *IEEE Computer System*, vol. 25, no. 3, pp. 38 – 49, March 1992.

[13] R. Culmone, G. Rossi, and E. Merelli, "An ontology similarity algorithm for bioagent," in *NETTAB Workshop on Agents nd Bioinformatics*, Bologna, 2002.

[14] B. Re, "Un modello di qualità per la scelta di servizi web in ambito biologico - il ruolo del modello di coordinazione," Master's thesis, Laurea in Informatica, Università di Camerino, a.y. 2003-2004, http://dmi.unicam.it/merelli/tesicl26/re.pdf.

[15] C. Ercoli, "Un modello di qualità per la scelta di servizi web in ambito biologico - il ruolo del middleware," Master's thesis, Laurea in Informatica, Università di Camerino, a.y. 2003-2004, http://dmi.unicam.it/merelli/tesicl26/ercoli.pdf.

[16] M. Wilkinson, D. Gessler, A. Farmer, and L. Stein, "The biomoby project explores open-source, simple, extensible protocols for enabling biological database interoperability," 2003, procedings of the virtual conference on genomics and bioinormatics. (3):17-27.

[5] http://www.owl-s.org
[6] http://www.discoveryLink.ibm.com
[7] http://www.mygrid.org
[8] http://hermes.cs.unicam.it