# Ontologies, matchmaking intermediation and interaction protocols in MASIR

Alma Gómez, Fco. J. Rodríguez, Pedro Cuesta, Rosalía Laza and Reyes Pavón

Dept. of Computer Science, University of Vigo
Campus As Lagoas, s/n, Ourense, 32004 Spain
{alma, franjrm, pcuesta, rlaza, pavon }@uvigo.es

**Abstract.** This work presents a prototype for information retrieval from heterogeneous sources, called MASIR. The system is composed of a set of agents which cooperate to return the relevant available information. The most significant issues presented are the use of ontologies for dealing with the heterogeneity of the different sources, the intermediation among the agents to achieve their goals and the interaction protocols among them. The comparison of MASIR with other relevant multiagent systems allows concluding that it can be useful in the environment described.

## 1 Introduction

Nowadays, the advances in Information Technology and Communications and particularly the Internet revolution provide a way of accessing to many sites trying to find the information needed by the user. This process implies an important set of problems like dealing with the distributed and heterogeneous nature of information or discovering the useful information from all entries returned.

In an environment as the described before, intelligent agents constitute a key technology for dealing with the problems introduced. Agents may assist the user in finding useful and relevant information, manage and overcome the difficulties associated with "information overload" [7], inform the user that new relevant data have been published, etc.

In this work the system MASIR (MultiAgent System for Information Retrieval), which is an example of the use of cooperative agents in an information retrieval environment is introduced. The paper is organised as follows. Section 2 introduces the theoretical aspects of intelligent information agents. Section 3 focuses in the description of MASIR prototype, addressing its architecture, interaction protocols and the use of ontologies. In section 4, other important multiagent systems are described and compared with MASIR. Finally, in section 5 conclusions and future work are presented.

# 2 Intelligent agents

In order to solve complex problems in heterogeneous environments, individual information agents must cooperate with other agents [5]. From this cooperation point of view (the agents may collaborate with other agents at the execution of tasks) we can distinguish two kinds of agents: cooperative and non-cooperative agents.

A set of cooperative agents which communicate to execute a task constitute a MultiAgent System (MAS). This cooperation allows solving problems that are beyond the individual capabilities of each individual information agent. Basically, the advantages of an approach based on cooperative agents are: simplicity, flexibility, robustness, scalability and the integration of existing legacy systems [11].

## 2.1 Non-cooperative agents

Enterprises and investigation groups have development a high number of non-cooperative information agents. We have distinguished several kinds of non-cooperative agents:

- *Search agents* (Bullseye, Copernic) that help the user to retrieve information from a close list of heterogeneous and distributed sources.
- *Monitor agents* (Mind-It, Informant) control the changes in different information sources (for instance, changes in web structures, updated news in a newspaper…).
- *Filter agents* (InfoScan, BotBox news) reduce the incoming information keeping only the relevant data from the user point of view.
- *Browser agents* (Interquick, Letizia [8]) help the user in navigation through the Web.
- *Agents for electronic commerce* (MySimon, Pricerunner) offer commercial services in order to save time and money.

## 2.2 Cooperative MAS

The cooperation determines the agent behaviour, this means that it closely defines the principal characteristics of the system design. Therefore, the complexity of system, and its functionality, depends on whether it has agents that make tasks without the collaboration of other agents (without communication) or it has several of them making tasks co-ordinately [9].

The previous idea shows that building a cooperative system is very complex because it is necessary to define its interactions and to implement protocols and methods of cooperation, like tasks delegation, contracts or negotiation between the autonomous agents that conform the system.

Furthermore, current investigation related to cooperative information agents is focussed on the two main functionalities that must be provided for the correct functioning of the whole system: a mechanism for linking the different agents and in the other hand a way of solving the heterogeneity of the information managed.

### 2.2.1 Intermediation

Taking into account the role played by the agents, two main kinds of agents can be distinguished [6]: *Provider agents* (Servers) offer their capabilities to users and other agents and *Requester agents* (Clients) use information and services offered by provider agents.

In a very simple multiagent system, the easiest method of coordination among these agents is the use of agent-to-agent messages. To make possible this coordination method, all providers and its available services have to be known in advance by every agent.

If the multiagent system is desired to be open (components/participants may enter and exit from the system at any time) this knowledge will not be managed by every agent. In conclusion, the problem of finding agents who might have, or produce, the information or other services needed by requesters is a complex problem.

Certainly, in an open multiagent information system it is necessary the presence of especial agents that mediate between providers and requesters. This kind of agents is called *Middle agents* and, as it has been said, mediates for a correct communication between providers and requesters.

The process of mediation done by middle agents is based on the following steps: (1) Provider agents advertise their capabilities to one or more middle agents, describing the service they provide. (2) Middle agents store all these advertisement. (3) A requester agent asks for locating and connecting to provider agents, which offer a desired service. (4) Middle agents, using the stored advertisements, return the result.

Depending on the kind of result returned, two types of middle agents may be distinguished:

- *Matchmaker agent*. The result is an ordered list of provider agents, which offer the requested service. Once this result is received by requester agent, the requester agent is the responsible for contacting the provider agent, negotiating and performing the transaction.
- *Broker agent*, in contrast with matchmaker, performs the complete transaction. This means that there is no direct communication between provider and requester agent, because all operations go through broker agent. The main tasks of these agents are contacting to appropriate provider agents, negotiating, performing and controlling the transaction and giving back the results of service to requester agent.

Given the fact that different types of middle agents provide different performance results, deciding what types of middle agents are appropriate depends on the application [6]. In addition, deciding between the use of matchmaking or brokering techniques to solve the connection problem yields to performance tradeoffs along a number of dimensions, both quantitative (such as the time needed to fulfil a request) and qualitative (such as the robustness and adaptivity of the system to the failure or addition of agents).

### 2.2.2 Ontologies

As it has been said, one of the main problems for cooperation in an agents society is the semantic heterogeneity of information that the agents must manage. Semantic heterogeneity considers the content of an information item and its “intended” meaning. In order to manage this semantic heterogeneity, the meaning of the interchanged information has to be understood across the agents society. This task can be done with the use of ontologies.

An ontology is defined as an explicit specification of a conceptualisation [4], that is, a representation (with a set of concepts and the relationships among them) of an abstract and simplified view of the world.

In agents society, ontologies can be used to describe the semantics of the request and service descriptions and to make explicit the content of the different information sources. They also reduce conceptual and terminological conflicts providing a unified framework.

Middle agent overcomes the semantic heterogeneity by means of a knowledge-based process, which relays on using ontologies. That is, the use of ontologies enables shared understanding among different agents with different aims and different viewpoints of the global system.

#### 2.2.3 Interaction protocols

Ongoing conversations between agents often fall into typical patterns. These typical patterns of message exchange are called interaction protocols [3]. Many standard interaction protocols have been defined and the ones relevant to MASIR will be introduced later.

## 3 MASIR architecture

The prototype presented is called MASIR (MultiAgent System for Information Retrieval). Its main aim is to provide uniform access to a set of heterogeneous sources of information, which in this case are documental databases. Based in a set of cooperative agents, the system will allow the user to make a request, which will be distributed among the available databases in order to achieve the results.

Figure 1 presents the system architecture which is composed of several kinds of agents. The characterisation of each agent is introduced below:

- *Interface agent*. Each user interacts with the system through its own interface agent. It is responsible for showing the user a request interface for entering its request using a boundary natural language taking into account the information contained in global ontology agent. Interface agent captures the request and translates and sends it to intermediary agent. Besides it presents to the user the obtained results and facilitates navigation through collection of retrieved documents. Interface agent shows this information using a user domain specific interface.
- *Ontology Agent*. Ontology agents are used for dealing with the concepts of the system. Two kinds of ontology agents can be distinguished: global ontology agent and local ontology agent. Global ontology agent provides the real world concepts and the semantic relationships among them, over which the user can make the request. Each source accessible in the system has its own local ontology agent which relates each real world concept with its particular representation in the data source.

- *Intermediary agent*. It is the core of the interrogation to databases process. It receives the request from interface agent, asks the matchmaker for the suitable wrappers, integrates the different results from the sources and gives them back to the interface agent.
- *Matchmaker agent*. Matchmaker agent works as a yellow pages service, giving back to the intermediary agent the list of wrapper agents it has to contact. The matchmaker agent consists of a list of concepts in the global ontology related to the wrappers which are able to provide information about each concept.

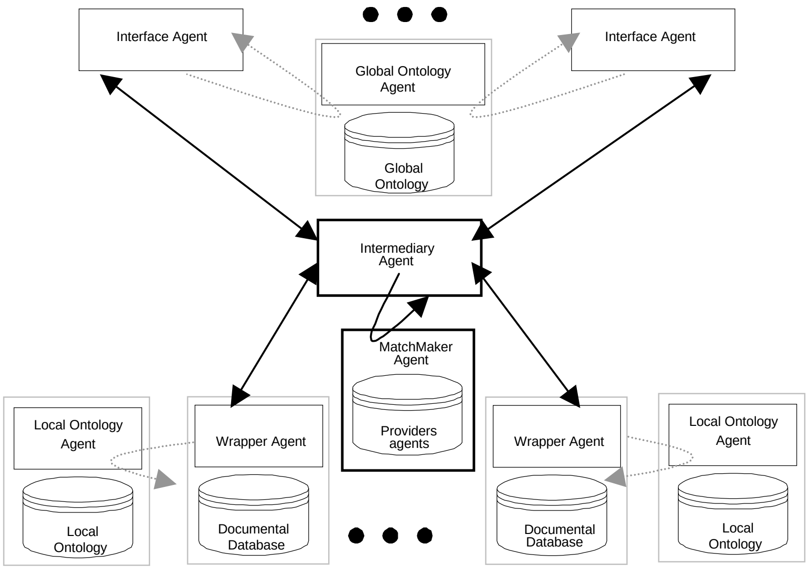


**Fig. 1.** MASIR Architecture.

- *Wrapper agent*. There is a wrapper agent for each information source. Wrapper agent knows the specific features of its related source, and is responsible for translating the subrequest received from intermediary agent in order to adjust it to its associated source of information and obtain the required data. Each request may imply, for example, the execution of SQL or OQL sentences or the utilisation of text retrieval techniques. In addition, wrapper agent accesses its source, adapts the results and sends them to intermediary agent. When a new source is added, its wrapper agent and its local ontology agent are prepared manually. The information introduced by this process will be used by wrapper agent in order to translate the request content to the particular schema of the source and its interrogation language.

### 3.1 Intermediation process

In MASIR provider agents are wrapper agents which offer their capabilities to interface agents (clients or requesters). Nevertheless some middle agents are introduce to mediate between wrappers and interface agents. These middle agents are intermediary agent and matchmaker agent.

The process of mediation implies the interactions described next (each interaction is numbered for its later reference):

- *(I1) Interface-Intermediary*. Intermediary agent accepts the request from interface agent. It analyses the request to isolate the concepts implied.
- *(I2) Intermediary-Matchmaker*. After detecting the main concepts intermediary agent asks matchmaker for a list of the wrapper agents with sources containing these concepts.
- *(I3) Intermediary-Wrapper*. Once intermediary agent has the list of wrappers it divides the request in subrequests (it may be the case that some sources were only able to solve part of the request, but not the whole) and distributes each subrequest to the appropriate wrapper. The division of request in subrequests is not always necessary and depends on the concepts contained in each source (this information is provided by matchmaker agent).
- *Wrapper-Intermediary*. The answers obtained from the different wrapper agents are integrated by intermediary agent before giving them back to the interface agent. This process may be not so straightforward as merging the results and may require, in some cases, some processing by the intermediary agent. This interaction is the answer in step number (I3).
- *Intermediary-Interface*. The results integrated in the previous step are sent to the interface agent which has begun the request process. This interaction is part of number (I1).

### 3.2 Ontologies

Ontology agents are used to deal with semantic heterogeneity among sources. They work with the concepts of the system which represent a simplified view of the information contained in the sources. In order to achieve this objective two kinds of ontology agents can be distinguished in MASIR system: global ontology agent and local ontology agent.

Global ontology agent has a conceptual model at higher abstraction level than the data models from each of the participant sources. This means that it provides concepts over which the user can make the request, and the semantic relationships among concepts.

In addition each source has its own local ontology agent. This agent relates each concept defined in the global ontology to its particular representation in the data source.

These agents collaborate with the rest of the system agents for solving a user request in the way explained next:

– *(O1)Interface-Global ontology*. The user interface must be dynamically generated for each user and the set of available information. For achieving this objective, interface agent should know what the request domain is. This information is obtained from global ontology agent, and it is used to help the user to specify the request.

– *(O2)Wrapper-Local ontology*. Wrapper agent asks local ontology in order to translate each concept contained in the subrequest received from intermediary agent to the corresponding concept in the source.

When a new source of information is added or removed from the system a process of actualization is needed. This process implies a first step which is done manually and consists of defining the local ontology and wrapper associated to the source. Next, the local ontology agent created communicates to global ontology agent the concepts the source is able to deal with and to matchmaker the list of concepts associated with the source.

### 3.3 Interaction protocols

Figure 2 is a sequence diagram which represents the underlying interaction protocols of the steps of the intermediation process (see section 3.1) and the ontology query process (section 3.2). Each step implies one or more interaction protocols which are characterised in the graph with a rectangle labelled with the same name.

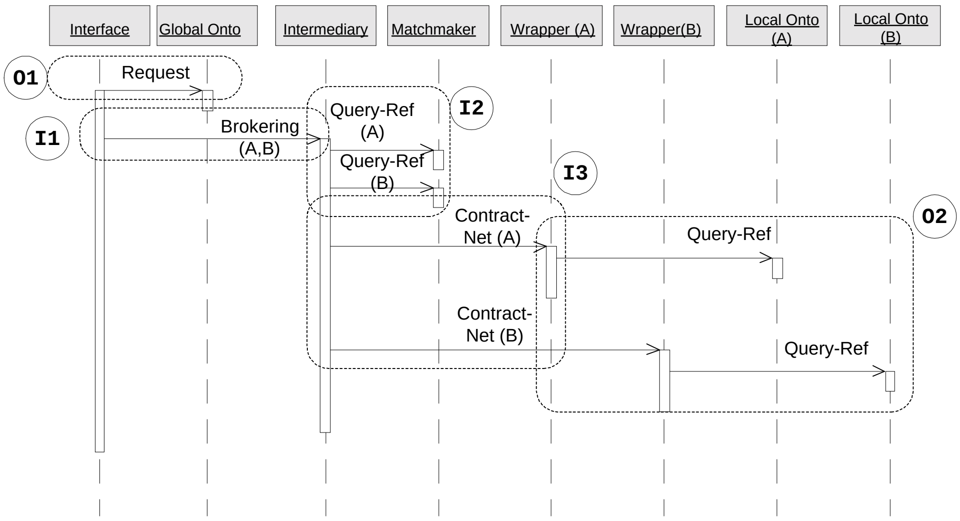


**Fig. 2.** FIPA interaction protocols used in the system.

Each box at the top represents a set of agents with the same role or functionality. For instance, interface box represents all interface agents in the system while intermediary box stands for the unique intermediary agent present in the actual configuration. A special case is the wrapper and local ontology agents because they are subdivided in two boxes in order to emphasize that their (wrapper and local ontology agents) communications are dependent on their underlying sources and related knowledge. Each arrow indicates a FIPA interaction protocol, that is, it represents the start of the conversation, the subsequent control messages and the final response [2,3].

In this example the user makes a request about two different sets of concepts, represented by A and B. This question implies the following interaction protocols:

- *O1*. The interface agent *speaks* with the global ontology agent using the FIPA Request Protocol. This interaction protocol allows the interface agent (initiator) to request the global ontology (participant) to perform the provide action of the domain knowledge.
- *I1*. With the FIPA Brokering Interaction Protocol the interface agent translates the user request (about A and B) to intermediary agent. The intermediary will determine a set of appropriate wrapper agents (helped by the matchmaker) for answering the request, will send the request (or sub-requests) to those agents and will give their answers back to the interface agent.
- *I2*. The intermediary agent asks the matchmaker about the relevant wrappers for a specific user request (A or B). They use for this task the FIPA Query-Ref interaction protocol.
- *I3*. When the intermediary agent knows the list of wrapper agents, negotiation between them starts (FIPA Contract-Net Interaction Protocol). This interaction will take place with different groups of wrapper agents depending on the concepts that they manage. The variable under negotiation is the units of time needed for obtaining a response (if a wrapper agent can not answer in a certain time, it will not be requested).
- *O2*. Each wrapper agent, in order to obtain the information of its underlying source, asks the local ontology agent with a FIPA Query-Ref interaction protocol about the requested concepts.

## 4 Related work

Some architectures have been proposed to construct MultiAgent Systems. In this point, we address the comparison of three of them with respect to MASIR. The basic issues which have been taken into account are related with intermediation and ontological aspects.

IMPACT (Interactive Maryland Platform for Agents Collaborating Together) provides a platform and environment for agent and software interoperability [10]. In this architecture, communication among agents and ontological services are centralized in a single agent called IMPACT Server. The process of intermediation is based on a yellow pages service, that is, matchmaking. Agents register their capabilities to IMPACT Server whenever they are created or when their capabilities are modified. With respect to ontological services in IMPACT, they seem to be less evolved providing only thesaurus and what they called Ontology/Translation Services.

RETSINA is an open multiagent system, which performs an information retrieval and integration directed by goals, in support of a variety of decision-making tasks [9]. The main concept underlying this environment is that relations among agents must be peer to peer, rather than result of a constraint in the architecture. The process of mediation basically relies on service matchmaking, provided by one or more middle agents. In RETSINA, agents advertise their capabilities using LARKS (Language for

Advertisement and Request for Knowledge Sharing), a specific language defined with the features of expressiveness, inference and ease of use. An agent looking for a service makes a request using LARKS and sends it to a middle agent. The process of matchmaking consists in comparing this request with the advertisements stored using different filters, which can be selected by user.

RETSINA allows the use of different local ontologies, each of them defined in the concept language ITL or by using WordNet. The matchmaker determines the relationships between two semantic descriptions computing the subsumption relation, which is related with the logical concept of implication.

Finally, InfoSleuth is an agent-based system that can be configured to perform many different information management activities in a distributed environment [1]. It is composed of seven kinds of software agents which, all together, provide a number of complex query services which involve solving ontology-based queries over the dynamically changing resources. Brokering agents are responsible for intermediation in this framework, making, in fact, a matchmaking process. One of the most outstanding features of InfoSleuth is its multibrokering architecture. A broker agent may store advertisements from requester or provider agents but also from other brokers. This lies to a complex system where great effort has been done in guaranteeing the intercommunication among all defined agents (avoiding islands of agents interconnected but isolated from the rest) and testing the performance of the overall system. On the other hand, due to multibrokering, the robustness, flexibility and performance is improved. With respect to ontologies, InfoSleuth uses the ontology agent which serves the set of ontologies supported by the application and can provide details of the ontology upon demand.

The system MASIR presented in this work follows many of the ideas presented in the related work. Particularly, the area of application is close to InfoSleuth system, nevertheless, multibrokering have not been considered necessary, because the number of sources is not high enough. Multibrokering, on the other hand would provide some kind of redundancy improving the robustness of the system, which will be an interesting feature.

MASIR does not incorporate a complex matchmaking process like RETSINA because the only information needed is the list of sources. The ontological necessities are also simpler than in RETSINA although the advantages of local ontologies are also relevant for our system.

## 5 Conclusions and future work

Many advantages in using cooperative agents for information retrieval in heterogeneous environments are noticed, specially simplicity, robustness and scalability. This work emphasizes three important aspects that must be taken into account in the construction of cooperative agents systems: the model of intermediation, the interaction protocols among the different agents and how ontologies are used to deal with the heterogeneity of information.

MASIR, the multiagent system proposed, uses two middle agents, one of them does a matchmaking process, while the other, intermediary agent, operates as a broker taking part into the request process. In order to solve semantic conflicts among the sources of information two kind of ontology agents are introduced.

The system is able to grow. When a new source is added, an associated wrapper agent and a local ontology agent are created and the global ontology agent is actualised incorporating the new semantics. At the moment this process is done manually by a human with a deep knowledge of the source. For future research in the domain of ontology, we propose to study in depth the way of automating the process for the actualisation of global ontology and matchmaker agents. This process will take into account the information available from the local ontology agent and will be done using the features of a description logic system.

In order to guarantee the robustness of the system, new analysis must be done about the possibility of replicating intermediary, matchmaker and global ontology agents for improving the availability of the system. This copy may also increase the performance of the system with a high number of sources.

Other important line of research is the definition of methodological aspects in the development. As it can be: description of agents and roles, association of interactions and roles, diagrams of different levels of abstraction, description of knowledge transference, etc.

## References


1. Bayardo, R. J.: Infosleuth: Agent-based semantic integration of information in open and dynamic environments. In ACM SIGMOD. (1997) 195-206
2. FIPA organization. http://www.fipa.org
3. FIPA Interaction Protocol Lib. Specification. http://www.fipa.org/specs/fipa00025/
4. Gruber, T.: Toward Principles for the Design of Ontologies Used for Knowledge Sharing. In International Journal on Human Computer Systems, 43 (1994) 907-928
5. Jennings, N.R..: Agent-Based Computing: Promise and Perils. Proceedings of the 16th International Joint Conference on Artificial Intelligence (1999) 1429-1436
6. Klusch, M. & Sycara K. Brokering and Matchmaking for Coordination of Agent Societies: A survey. In A. Omicini (eds): Coordination of Internet Agents. Models, Technologies, and Applications. Berlin: Springer-Verlag., (2001) 197-224
7. Klusch, M.: Intelligent Information Agents -Agent-Based Information Discovery and Management on the Internet. Berlin Springer-Verlag (1999)
8. Lieberman, H.: Letizia: An Agent That Assists Web Browsing. In Proceedings of the 14th International Joint Conference on Artificial Intelligence. Morgan Kaufmann (1995) 924-929
9. Sycara, K.,Paolucci, M., van Velsen, M., & Giampapa, J. The RETSINA MAS Infrastructure. Robotics Institute Technical Report # CMU-RI-TR-01-05. (2001)
10. T.J. Rogers, Robert Ross and V.S.Subrahmanian.: IMPACT: A system for building agent applications. (eds): Journal of Intelligent Information Systems. Kluwer Academic Publishers. (1996) 275-294
11. Wooldridge, M & Jennings, N.R. Intelligent Agents: Theory and Practice. The Knowledge Engineering Review 10(2) (1995) 115-152