AGENTS SUPPORTING INFORMATION INTEGRATION: THE MIKS FRAMEWORK

G. Gelati, F. Guerra, M. Vincini

University of Modena and Reggio Emilia
DSI, Via Vignolese 905, 41100 Modena, Italy

{gelati,guerra,vincini}@dsi.unimo.it

Abstract

During past years we have developed the MOMIS (Mediator envirOnment for Multiple Information Sources) system for the integration of data from structured and semi-structured data sources.

In this paper we propose a new system, MIKS (Mediator agent for Integration of Knowledge Sources), which enriches the MOMIS architecture exploiting the intelligent and mobile agent features.

1. Motivation

The web explosion, both at Internet and intranet level, has transformed the electronic information system from single isolated node to an entry point into a worldwide network of information exchange and business transactions. One of the main challenges for the designers of the e-commerce infrastructures is the information sharing, retrieving data located in different sources thus obtaining an integrated view to overcome any contradiction or redundancy.
During past years we have developed the MOMIS (Mediator envirOnment for Multiple Information Sources) system for the integration of data from structured and semi-structured data sources.

In this paper we propose a new system, MIKS (Mediator agent for Integration of Knowledge Sources), which enriches the MOMIS architecture exploiting the intelligent and mobile agent features.

2. The MOMIS System

The **MOMIS** system (Mediator envirOnment for Multiple Information Sources) [2,3,4] aims to integrate data from structured and semi-structured data sources. The MOMIS system supports the designer in semi-automatic integration of heterogeneous sources schema (relational, object, XML and semi-structured sources). By using the MOMIS system, the designer may carry out integration following a semantic approach that uses intelligent OLCD Description logics-based techniques, clustering techniques and an ODM-ODMG extended model to represent extracted and integrated information, ODM_{I3}. Using the ODL_{I3} language, referred to the ODM_{I3} model, it is possible to describe the sources (local schemata) and perform an integrated view of all the sources, i.e. Global Virtual View (GVV), which is expressed using XML standard. The use of XML in the definition of the Global Virtual View permits to export the MOMIS metadata to other local and GVV mediators by the interchange of XML data files.

Like other integration projects [1,8], MOMIS follows a "semantic approach" to information integration based on the conceptual schema, or metadata, of the information sources, and on the the I^3 architecture [6] (see figure 1). The system is composed by the following functional elements that communicates using the CORBA (OMG) standard:

1. a common data model, ODM_{I3}, which is defined according to the ODL_{I3} language, to describe source schemas for integration purposes. ODM_{I3} and ODL_{I3} have been defined in MOMIS as subset of the corresponding ones in ODMG, following the proposal for a standard mediator language developed by the I^3/POB working group [5]. In addition, ODL_{I3} introduces new constructors to support the semantic integration process;
2. *Wrappers*, placed over each sources, translate metadata descriptions of the sources into the

common ODL_{I3} representation, translate (reformulate) a global query expressed in the OQL_{I3}[1] query language into queries expressed in the sources languages and export query result data set;

3. *Mediator*, which is composed of two modules: the *SI-Designer* and the *Query Manager* (QM). The SI-Designer module processes and integrates ODL_{I3} descriptions received from wrappers to derive the integrated representation of the information sources. The QM module performs query processing and optimization. The QM generates OQL_{I3} queries to be sent to wrappers starting from each query posed by the user on the Global Schema. QM automatically generates the translation of the query into a corresponding set of sub-queries for the sources and synthesizes a unified global answer for the user.

MOMIS provides the capability of explicitly introducing many kinds of knowledge for integration, such as integrity constraints, intra- and inter-source intensional and extensional relationships, and designer supplied domain knowledge. A *Common Thesaurus*, which has the role of a shared ontology of the sources, is built in a semi-automatic way. The *Common Thesaurus* is a set of intra and inter-schema intensional and extensional relationships, describing inter-schema knowledge about classes and attributes of sources schemas; it provides a reference on which to base the identification of classes candidate to integration and subsequent derivation of their global representation.

MOMIS supports information integration in the creation of an integrated view of all sources (Global Virtual View) in a way automated as much as possible and performs revision and validation of the various kinds of knowledge used for the integration. To this end, MOMIS combines reasoning capabilities of Description Logics with affinity-based clustering techniques, by exploiting a common ontology for the sources constructed using lexical knowledge from WordNet [7] and validated integration knowledge.

The Global Virtual View is expressed by using XML standard, to guarantee the interoperability with other open integration systems.

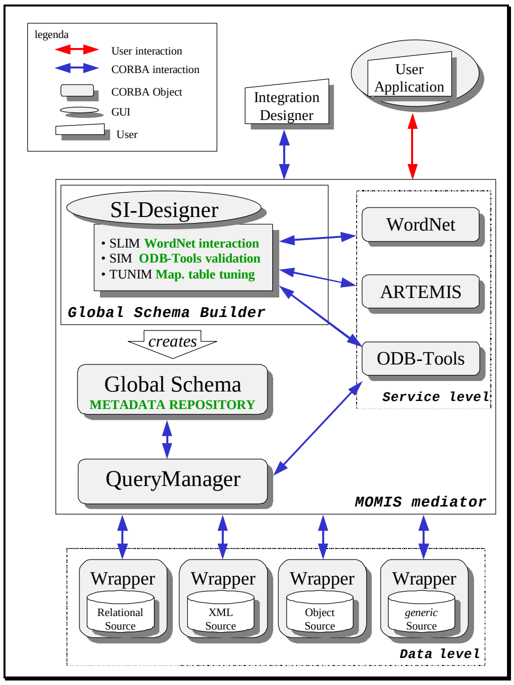

Figure1 – The MOMIS system architecture

3. Introducing Agents to support the MIKS' integration process

The MOMIS implementation is based on object-oriented technology. A feasible development we have been taking into account is to introduce the agent technology. Basically, the exploitation of agents could help in improving some of the existing MOMIS features and show up possible new functionalities we could add to the system. The new architecture of our system, named MIKS system, is shown in Figure 2.

Being an integration tool, the MIKS system relies in general on a set of remote sources and performs some analysis and manipulation of their contents. At present, this kind of communication and distributed tasks is carried out by means of CORBA. One feasible improvement is to introduce intelligent and mobile agents in order to accomplish those tasks in a more efficient and flexible way. A few environments suitable for developing multi-agent systems are JADE [9], ObjectSpace's Voyager [12], IBM's Aglets [11], Matchmaking [10]. The exploitation of agents can be particularly useful with regard to the whole integration process: the search phase and the query rewriting phase.

[1] OQL_{I3} is a subset of OQL-ODMG.

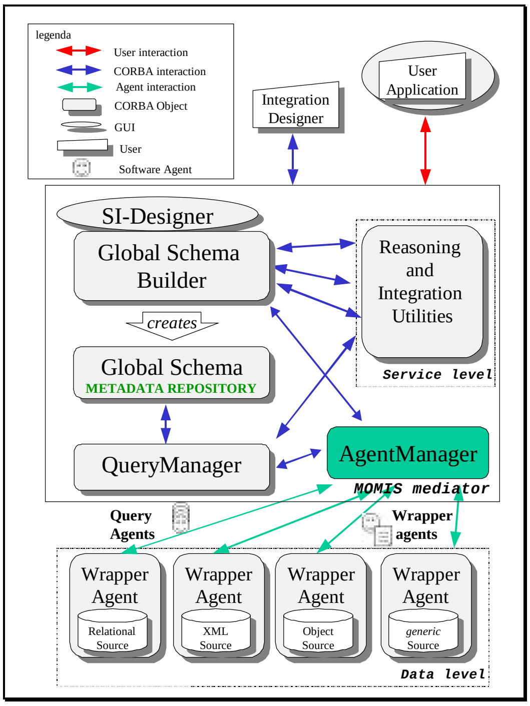

Figure 2 - The MIKS system architecture

3.1. Searching for desirable sources

The set of remote sources to be integrated into the MIKS global view can be made up by sources autonomously deciding to provide to the system their knowledge. On the other hand, at any moment a need to increase the number of the sources may arise in order to enhance the quantity and quality of available data. In such a case, the system has to search for new potentially interesting sources in a certain domain (e.g. Internet). The search may exploit in this case intelligent and mobile agents. The gray box depicted in Figure 2 shows the agent framework we have designed. As we can see, the Agent Manager (AM) acts as a centralized facility to handle the whole agent life cycle. The first step to be undertaken whenever a new search for potential clients has to be executed is creating and coordinating a set of agents visiting interesting sources. The list of interesting sources can be provided from the system designer who starts the search and/or can be obtained by a domain search according to the ontology contained into the Common Thesaurus. The AM instantiates the set of agents and allot the addresses of the list among the agents. Inside the group, one agent is recognized as director while the others play the role of explorers. One agent will be sent to one source at a time. The **Mobile agents** explorer then visit the assigned sources and, once arrived at the address, they have to introduce themselves, clarify the goal of their visit and negotiate the availability of the source to provide access to a part or the whole of its content. After having completed this task, the agent come back to AM and communicates the result of the search (successful or unsuccessful) and whether new potentially interesting sources were pointed out by the visited source to the director. These new links will be dynamically added to the director's list and the new addresses will be assigned to some agents, following the usual method. This way, while the search process progresses, the coordinator keeps track of the global state of the search and can decide when it is better to stop.

If the result of the negotiation phase is positive, a second step is required to satisfy the original need for new data. Precisely, the source has to be integrated and therefore the director notifies the AM that an **intelligent agent** has to be sent as a wrapper to the source.

Finally, it is clear that for a given source the two identified phases have to be carried out following a chronological order, but in general during the search process the two steps have not to comply with a sequential order, as an explorer can be sent to source "A" while a wrapper is invoked after a successful negotiation for "B".

3.2. Supporting information retrieval

The second phase in which we apply intelligent agents is query rewriting. The Query Manager (QM) is charged to perform query processing: whenever a MIKS user poses a query over the integrated global view, it generates in an automatic way the translation of the posed global query into different sub-queries, one for each involved local source. The wrapper located on the source site then carries out the sub-query and the result is propagated to the MIKS system. If many queries or heavy queries have to be answered then the MIKS system should be able to handle the required number of connections and the communication cost could rapidly increase due to the amount of data to be transferred. In order to avoid a heavy workload to the system and to the network, each sub-query (or a subset or the whole set) can be instead assigned to a query agent. The query agent will be sent to the source site where it carries out the search computation locally, interacting with the wrapper agent. Consequently, a smaller amount of data has to be propagated to the MIKS system. A second benefit is that the query has not to be solved straight away, but, since the query agent is on the source site, the MIKS system could even retrieve the result with some delay if necessary (e.g. when the connection is unstable or if a system has no permanent connection).

4. References

[1] Y. Arens, C.Y. Chee, C. Hsu, and C. A. Knoblock, 1993. ‘Retrieving and integrating data from multiple information sources’. *International Journal of Intelligent and Cooperative Information Systems, 2(2):127–158, 1993*.

[2] D. Beneventano, S. Bergamaschi, S. Castano, A. Corni, R. Guidetti, G. Malvezzi, M. Melchiori, and M. Vincini, 2000. ‘Information integration: The MOMIS project demonstration’. *Proceedings of 26th International Conference on Very Large Data Bases, September 10-14, 2000, Cairo, Egypt, pages 611–614. Morgan Kaufmann, 2000*.

[3] S. Bergamaschi, S. Castano, and M. Vincini, 1999. ‘Semantic integration of semi-structured and structured data sources’. *SIGMOD Records, 28(1), March 1999.*

[4] S. Bergamaschi, S. Castano, D. Beneventano, and M. Vincini, 2001. ‘Semantic integration of heterogenous information sources’. *Journal of Data and Knowledge Engineering, 36(3):215–249, 2001*.

[5] P. Buneman, L. Raschid, and J. Ullman, 1996. ‘Mediator languages - a proposal for a standard’, *April 1996. Available at ftp://ftp.umiacs.umd.edu/pub/ONRrept/-medmodel96.ps.*

[6] R. Hull and R. King et al., 1995. ‘Arpa I^3 reference architecture’, 1995.
Available at http://www.isse.gmu.edu/I3Arch/index.html.

[7] A. G. Miller, 1995. ‘Wordnet: A lexical database for English’, *Communications of the ACM, 38(11):39–41, 1995.*

[8] M. T. Roth and P. Scharz, 1997. ‘Don’t scrap it, wrap it! a wrapper architecture for legacy data sources’. *In Proc. of the 23rd Int. Conf. on Very Large Databases, Athens, Greece, 1997*.

[9] F.Bellifemine, A.Poggi, G.Rimassa, P. Turci, “An Object-oriented framework to Realize Agent Systems”

[10] K. Sykara and M. Klusch and S. Widoff and J. Lu, 1999. “Dynamic Service Matchmaking Among Agents in Open Information Environments”, *SIGMOD Records, 28(1), March 1999.*

[11] IBMAglets,
http://www.ibm.com/aglets

[12] ObjectspaceVoyager,
http://www.objectspace.com