The SEWASIE MAS for Semantic Search

D. Beneventano, S. Bergamaschi, F. Guerra, M. Vincini
University of Modena and Reggio Emilia, Italy
beneventano.domenico,bergamaschi.sonia,guerra.francesco,vincini.maurizio@unimore.it

1. Introduction

The capillary diffusion of the Internet has made available access to an overwhelming amount of data, allowing users having benefit of vast information. However, information is not really directly available: internet data are heterogeneous and spread over different places, with several duplications, and inconsistencies. The integration of such heterogeneous inconsistent data, with data reconciliation and data fusion techniques, may therefore represent a key activity enabling a more organized and semantically meaningful access to data sources. Some issues are to be solved concerning in particular the discovery and the explicit specification of the relationships between abstract data concepts and the need for data reliability in dynamic, constantly changing network. Ontologies provide a key mechanism for solving these challenges, but the web's dynamic nature leaves open the question of how to manage them.

Many solutions based on ontology creation by a mediator system have been proposed: a unified virtual view (the ontology) of the underlying data sources is obtained giving to the users a transparent access to the integrated data sources [1, 2, 3]. The centralized architecture of a mediator system presents several limitations, emphasized in the hidden web [4]: firstly, web data sources hold information according to their particular view of the matter, i.e. each of them uses a specific ontology to represent its data. Also, data sources are usually isolated, i.e. they do not share any topological information concerning the content or structure of other sources.

Our proposal is to develop a network of ontology-based mediator systems, where mediators are not isolated from each other and include tools for sharing and mapping their ontologies. In this paper, we describe the use of a multi-agent architecture to achieve and manage the mediators network. The functional architecture is composed of single peers (implemented as *mediator agents*) independently carrying out their own integration activities. Such agents may then exchange data and knowledge with other peers by means of specialized agents (called *brokering agents*) which provide a coherent access plan to the peer network. In this way, two layers are defined in the architecture: at the local level, peers maintain an integrated view of local sources; at the network level, agents maintain mappings among the different peers.

The result is the definition of a new type of mediator system network intended to operate in web economies, which we realized within SEWASIE (SEmantic Webs and AgentS in Integrated Economies), an RDT project supported by the 5th Framework IST program of the European Community, successfully ended on September 2005.

The paper is structured as follows: section 2 introduces the SEWASIE architecture. Section 3 describes the building process of the two-level SEWASIE ontologies. Section 4 describes the main SEWASIE agents. Finally, section 5 describes some related works and section 6 sketches out some conclusions.

2. SEWASIE architecture

The functional architecture of the SEWASIE system is based on a Multi Agent System (MAS) composed of a network of information (mediator) agents, which represent the peer and are called SINodes, and a set of agents to support users querying the underlying peers as a single transparent data source (see Figure 1).

The SEWASIE system was designed according to a coordination strategy based on task decomposition and distribution [5]. Task decomposition is based on the layout of the information resources and physical actors, as well as the expertise of available agents, while task distribution is based on an organizational structure where agents have fixed responsibilities for particular tasks. Thus, a specific and simple task to accomplish is delegated to each agent, avoiding the assignment of extreme computational burden. According to their role, the agents participating in the SEWASIE MAS may be organized in four different categories, namely information (mediator), querying, brokering and user agents. The user interacts with a web interface, managed by the User Agent, that provides the list of available information brokers (brokering agents) and allows the creation of queries over the ontology.

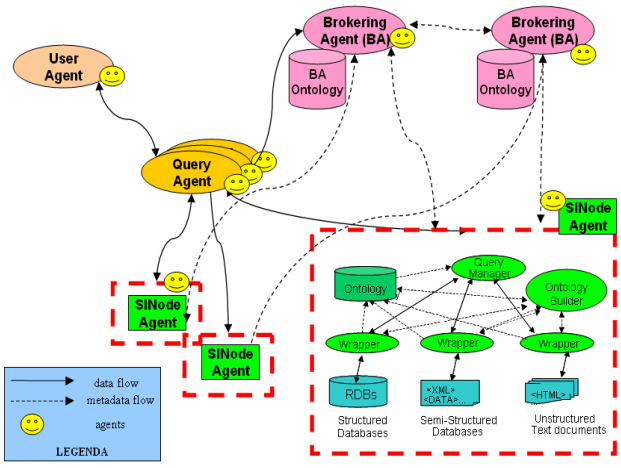

Fig. 1. SEWASIE Network Architecture.

The User Agent invokes the instantiation of a Query Agent that translates, by means of the Brokering Agents, the user request in a set of conjunctive queries that will be performed at the information (mediator) level. Finally, the Query Agent performs the fusion of the single answers and returns the data in XML format to the User Agent for the web visualization process. [6]
The SEWASIE Information Nodes (SINodes) consist of a specific Ontology, created by an Ontology Builder, and a Query Engine that executes queries over the Ontology, are the core of the SEWASIE system. The ontology holds a virtual view of the overall information managed within a SINode and includes a set of heterogeneous information sources, wrappers, and a metadata repository.
On top of the SINodes, a Brokering Agent network is created to maintain information of peers being members of the SEWASIE network, whether they are available at a given time to solve queries posed to the system or request to update the ontology.
Query Agents are the carriers of the user query from the user interface to the SINodes, and serve the purpose of solving a query by interacting with the brokering agent network. Once this process is over, all partial results are fused into a final answer to be delivered to the user.

A User Agent includes a web query tool that guides the user in composing queries. It is responsible for contacting brokering agents in order to get ontologies to be visualized and is also responsible of managing the set of query agents required to solve users' queries and to present the result data trough the web interface.

3. SEWASIE Ontology creation

The SEWASIE network provides two different ontology levels: the lower level, where SINode Ontologies represent groups of data sources with semantically close contents, and the upper level, where

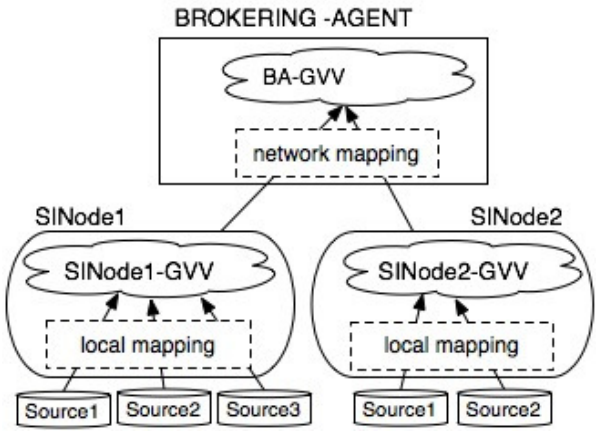

Fig. 2: The BA/SINodes ontologies and mappings

BA Ontologies represent SINodes with semantically close contents.
The relations between the two ontology levels are shown in Figure 2:

- an SINode contains a GAV[1] mediator-based data integration system, which integrates heterogeneous data sources into an ontology consisting of an *annotated*[2] Global Virtual View, denoted by SINode-GVV, and Mappings to the data source schemata.
- a Brokering Agent contains a mediator-based data integration system, which integrates the SINode-GVV of its peers into an ontology composed of an *annotated* Global Virtual View, denoted by BA-GVV, and Mappings to the SINode-GVVs.

From a theoretical point of view, the proposed architecture, based on two different levels of mappings, represent a non-traditional setting in data integration and an interesting case of mapping composition. In fact, [7] showed that, in general, the mapping from the sources to the BA-GVV is not simply the composition of local and network mapping (see figure 2); on the contrary, while in our case both local and network mapping are GAV mappings, it was proved that it is possible to consider a "global" mapping as the composition of local and network mapping. The semantics of the SEWASIE system, is defined in [8, 9].
As a domain example, we applied the methodology on

[1] In the Global-As-Vies (GAV) approach the contents of the elements of the Global Virtual View is not predefined and is described in terms of a view of the local sources.

[2] Annotation is the association of each element (attribute class) of a data source schema with one or more synset of Wordnet (see section 3.1)

the integration of four Web Italian sites containing information about enterprises and products in the Mould Mechanical domain. In particular, the Comitato Network Subfornitura Italian Web site (www.subfor.net) allows the users to query an online database (about 5,000 enterprises) where detailed information on Italian enterprises and their products can be found. The second website (www.plasticaitalia.com) is the "yellow pages" of Italian plastic companies (about 6,500). The third website (www.tuttostampi.com) collects about 4,000 Italians industrial moulding companies. Finally, we analyze a Web portal (www.deformazione.it) where about 2,500 Italian companies working on metallic sheets are presented.

3.1. GVV Generation Process

The GVV Generation process can be outlined in different steps.

Extraction of Local Source Schemata: Wrappers acquire schemata of the involved local sources and convert them into ODLI3[4]. Schema description of structured sources (e.g. relational database and object-oriented database) can be directly translated, while the extraction of schemata from semistructured sources need suitable techniques as described in [10]. To perform information extraction and integration from HTML pages, research and commercial web data extraction tools, such as ANDES [11], Lixto [12] and RoadRunner [13], have been experimented and adopted to wrap the selected web sites.

Local Source Annotation: Terms denoting schema elements in data sources are semantically annotated according to a common lexical reference in order to provide a shared meaning to each of them. We chose the WordNet database as lexical reference. The system automatically detects, for each term in the sources, the (most commonly) used meaning present in WordNet. Algorithm for automatic annotation prepares terms by applying stop-words and stemming functionalities to enhance the accuracy result. Then the Ontology Designer can manually revise the meaning(s) for each annotated term.

In our example, the recall rate of the terms automatically annotated in the sources is 77%, with a precision of 82%.

Common Thesaurus Generation: MOMIS builds a Common Thesaurus that describes intra and inter-schema knowledge in the form of: synonyms (SYN), broader terms/narrower terms (BT/NT), meronymy/holonymy (RT) relationships. The Common Thesaurus is incrementally built by starting from schema-derived relationships, i.e. automatic extraction of intra-schema relationships from each schema separately. Then, the relationships existing in the WordNet database between the annotated meanings are exploited by generating relationships between the respective elements that are called lexicon-derived relationships. The Ontology Designer may add new relationships to capture specific domain knowledge, and finally, by means of a Description Logics reasoner, ODB-Tools [14], which performs equivalence and subsumption computation) infers new relationships and computes the transitive closure of Common Thesaurus relationships.

In the example, 517 relationships are computed, 7% obtained by the schemata, 73% derived from WordNet relationships, none added by the designer and 20% obtained by inference and transitive closure.

GVV generation: Starting from the Common Thesaurus and the local sources schemata, MOMIS generates a GVV consisting of a set of global classes, plus mappings to connect the global attributes of each global class and the local sources' attributes. Going into details, the GVV generation is a process where ODLI3 classes describing the same or semantically related concepts in different sources are identified and clusterized in the same global class by means of the ARTEMIS tool [15]. Clusters for integration are interactively selected from the affinity tree using a non-predefined threshold based mechanism. The Ontology Designer may interactively refine and complete the proposed integration results. For example, the obtained SINode1's GVV contains the global classes Company, Country, Province, List_of_category, Category and a set of classes composing a (partial) hierarchy of the managed categories.

GVV annotation: The GVV is automatically annotated, i.e. each of its elements is associated to the broadest meanings extracted from the annotated sources. The GVV annotation can be useful to make the meaning of the created domain ontology understandable to external users and applications [15].

3.2. Query Unfolding

The query unfolding process is performed for each *Single Global Query Q* over a global class *C* of the *GVV* (for sake of simplicity, we consider the query in an SQL-like format):

Q = SELECT <Q_SELECT-list> from C where <Q_condition>

where *<Q_condition>* is a Boolean expression of positive atomic constraints: *(GA1 op value)* or *(GA1 op GA2)*, with *GA1* and *GA2* attributes of *C*. Let *L1,L2, . . . Ln be* the *local* classes related to the *C*, i.e. which are integrated into *C*.

Let us consider the SQL version of ExpAtom1:

SELECT Name,Address
FROM Company
WHERE Region = 'Veneto' and Capital_Stock > 50
The (portion of) the Mapping Table of the class Company involved in the query is:

Company	SN1.Enterprise	SN2.Company
Company_ID	Company_ID	Company_ID
Address	Address	Address
Capital_Stock	Capital_Stock	
Region	Region	Region
SubContractor		SubContractor

where
• the Join Condition is SN1.Enterprise.COMPANY_ID =SN2.Company.COMPANY_ID

• Subcontractor, Region and Capital_Stock are homogeneous attributes

• Address is defined by a precedence function.

The query unfolding process is made up of the following three steps:
Step 1) Generation of *Local Queries*:

For each EXPAtoms a set of local queries is generated. A local query, denoted by FDAtom, is a Single Local Class Query, i.e., a query on a single local class L:

FDAtom = SELECT <SELECT-list>
FROM L WHERE <condition>

where L is a local class related to the global class C.
The <SELECT-list> is computed by considering the union of:
the global attributes in <Q_SELECT-list> with a not null mapping in L,
the global attributes used to express the join conditions for L,
the global attributes in <Q_condition> with a not null mapping in L.
The set of global attributes is transformed in the corresponding set of local attributes according to the Mapping Table. The <condition> is computed by performing an atomic constraint mapping: each atomic constraint of <condition> is rewritten into one that is supported by the local source. The atomic constraint mapping is performed according to the Data Conversion Functions and Resolution Functions defined in the Mapping Table. For example, if the numerical global attribute GA is mapped onto L1 and L2, and we define AVG as the resolution function, the constraint (GA = value) cannot be pushed at the local sources, because AVG has to be calculated at a global level. In this case, the constraint is mapped as true in both the local sources. On the other hand, if GA is an homogeneous attribute the constraint can be pushed at the local sources. For example, an atomic constraint (GA op value) is mapped onto the local class L as follows:

(MTF [GA][L] op value) if MT [GA][L] is not null and the op operator is supported into L

true otherwise

The set of FDAtoms for Expatom1 is:

```
FDAtom1 = SELECT COMPANY_ID,NAME,
                 REGION,ADDRESS
          FROM SN1.company
          WHERE (REGION like 'VENETO')
FDAtom2 = SELECT
COMPANY_ID,NAME,REGION,ADDRESS
                 FROM SN2.company
                 WHERE ( REGION like
'VENETO' and CAPITAL_STOCK > 50 like 'yes')
```

Step 2) Generation of FDQuery which computes the Full Disjunction of the FDAtoms

In our example:
FDQuery = SELECT * FROM FDATOM1 full join FDATOM2 on (FDATOM1.Company_ID=FDATOM2.Company_ID)
Step 3) Generation of the final query (application of Resolution Functions): for Non-Homogeneous Attributes (e.g. Address) we apply the associated Resolution Function (in this case the precedence function).

4. The SEWASIE agents at work

SINodes expose their GVVs on the network and software agents act as a glue between the different peers. Peers are recognized as being part of the SEWASIE system as long as they register their GVVs by a brokering agent. From a deployment view point, the SEWASIE network is a multi-agent distributed system developed by using Java Agent DEvelopment (JADE) platform (jade.cselt.it). In this section we introduce the main tasks performed by the SEWASIE agents.

4.1. User Agents

UAs are in charge of all search activities in the SEWASIE network, they mainly have a coordination role. The main functions implemented in a UA are:

- find-initial-BA: the initial BA must be reached and, if available, contacted in order to get its ontology mappings.

- process-query: when the user has finished the query creation by means of the user interface, two tasks are accomplished: a new QA is created, and the query translated into the internal query language is passed within a message to the QA.
- receive-results: by means of this function, the results collected by a QA are received and transferred to the query tool in order to be shown to the user.

4.2. Query Agent

The Query Agents' goal is to define and execute the global query process strategy. This task is performed by means of two different processes involving the initial BA:

1. on the basis of the query and the Brokering Agent (BA) Ontology, the BA establishes a list of SINodes Agents which have to be contacted in order to solve the query;
2. new BAs selection: the initial BA is connected to other BAs, and provides the QA with a list of useful BAs to be contacted on the basis of the knowledge of the GVV of the near BAs.

The life cycle of a QA is initiated by an invocation of solve-query service, and it is finished when results are delivered. The following actions are combined in a QA to respond to a solve-query demand:

- validate-query: The agent must parse the query received from the UA in order to check whether it is well-defined, and to extract from it the information about the initial BA.
- query-BA: This action translates the query in terms of the BA GVV in order to have, as a response, a set of relevant SINode's queries and a set of additional BAs to be consulted.
- query-SINode: By means this action, the SINodes which are useful in order to answer the query are contacted, and all responses are merged into a single result.
- deliver-result: When the query process is finished, the results are returned to the UA by means of this action.

4.3. Brokering Agents

Brokering Agents are responsible for maintaining metadata about the SEWASIE network. These metadata are able to describe the ontologies of the underlying SINodes, along with information about the other near BAs.

The life cycle of a BA is initiated when an authorized user creates the BA and registers them to the Directory Facilitator service (DF). The DF is the standard JADE yellow-pages service: agents can advertise to the DF their capabilities and keep updated the information about their status. In this process, knowledge of existing SINodes and other BAs should be immediately handed over to the newly created BA, in order to build the local GVV. Being in the active state, a BA may receive messages for updating its knowledge, or for consulting its knowledge. When serving the former messages, the BA is said to be in design phase, while serving the latter the BA is in query phase.

The following actions can be performed by a BA:

- broadcast-ontology: This action is performed when the GVV of the BA is updated because of a newly added SINode. It involves deciding which other BA could be involved in the updated ontology, and its packaging and sending.
- query expansion: This action is performed when a query is submitted by a QA and a list of single queries for the relevant SINodes is created.
- find-relevant-BA: This action is performed when a query is submitted by a QA in order to have a set of other BAs which are useful for solving the query.
- deliver-answer: By means of this action, partial results from the previous two actions are collected and delivered to the QA.

4.4. SINodes Agents

SINode Agents group together several data sources, providing a logical node of information to the SEWASIE network. These nodes may be spread over several machines, and have significant resources allocated.

Once SINodes are created, they should be related to one or more BAs in order to belong to the SEWASIE network. The full list of available Bas is retrieved by querying the DF. The list of brokering agents may be further filtered according to selective parameters, such as the number of GVVs already integrated by a brokering agent or to its workload. From the point of view of agents, the SINode Agent plays two different roles: it contacts a BA and waits to be contacted by a QA. In the first case, the SINode Agent must send its GVV to the BA in order to be added to the BA GVV. In the second case, the QA sends to the SINode Agent a query that must be answered.

5. Related work

Several agent-based information retrieval systems have been developed. For comparison with similar systems, let us introduce the SEWASIE main characteristics which make the SEWASIE system unique among the agent-based information retrieval systems:

Two-level data integration schema: Strongly tied local nodes are integrated into SINodes; BAs provide globally integrated ontologies by means of weaker mappings.

Query management: Query building assisted by a query tool, query rewriting in the two levels of data integration following local ontologies using robust and complete algorithms.

CARROT II [16] is one of the most common systems: it is an agent-based architecture for distributed information retrieval and document collection management. It consists of an arbitrary number of agents providing search services over local document collections or information sources. They contain metadata describing their local documents which are sent to other agents that act as brokers. There are many differences between SEWASIE and CARROT: in the latter there is no support for the user in creating the query, and metadata information is not reflected in the process of query building. Moreover, CARROT agents only perform a routing of the query to relevant information sources, no query rewriting is done in this step. In SEWASIE, the query is reformulated following brokering agent's ontology before asking SINodes, which contain the information sources.

Several other information retrieval systems using routing agents are known, such as HARVEST [17], CORI [18] and InfoSleuth [19]. Other systems, like TSIMMIS [1], include some rewriting rules against predefined query patterns. There are several steps of query processing also in the MISSION project [20]. In these cases, data integration technology is not present or, as in TSIMMIS, is limited to automatic generation of wrappers and mediators from web pages. In SEWASIE, the data integration techniques [14] adopted by SINodes address unstructured and semi-structured data sources, as well as relational databases.

6. Conclusion

In this paper, we provided a general overview of the SEWASIE multi-agent architecture. We showed the different kinds of agents composing the system and how they are organized. By means of a running example we described the techniques implemented in SEWASIE for integrating and querying data sources by means of ontologies: test is made up of 3 Bas and 8 SINodes belonging to 30.000 records.

The SEWASIE project successful ended on 2005, but the research on these topics is continuing mainly in the field of developing techniques for executing queries in a p2p architecture.

7. References

[1] Garcia-Molina, H., Papakonstantinou, Y., Quass, D., Rajaraman, A., Sagiv, Y., Ullman, J. D., Vassalos, V., and Widom, J. 1997. The TSIMMIS Approach to Mediation: Data Models and Languages. J. Intell. Inf. Syst. 8(2).

[2] Kirk, T., Levy, A. Y., Sagiv, Y., Srivastava, D. 1995. The Information Manifold. In Knoblock, C., Levy, A. eds, Stanford.

[3] Bergamaschi, S., Castano, S., Beneventano, D., and Vincini, M. 2001. Retrieving grating data from multiple sources: the MOMIS approach. Data Knowl. Eng. 36.

[4] S. Raghavan, H. Garcia-Molina: Crawling the Hidden Web. VLDB 2001: 129-138.

[5] Weiss, G. 2000. Multigent Systems – A Modern Approach to Distributed Artificial Intelligence. Cambridge, MA: MIT Press, pp. 79-120.

[6] Beneventano D., Bergamaschi S. Semantic search engines based on data integration systems. In Semantic Web Services: Theory, Tools and Applications. Idea Group Publishing, 2006.

[7] Madhavan, J., and Halevy, A.Y., 2003. Composing Mappings Among Data Sources. In VLDB, (pp. 572-583).

[8] Calì, A., Calvanese, D., Di Giacomo, G. D., and Lenzerini, M. 2004. Data integration under integrity constraints. Inf. Syst., 29 (2), (pp. 147–163).

[9] Beneventano, D., and Lenzerini, M. 2005. Final release of the system prototype for query management. Sewasie, Deliverable D.3.5.

[10] Abiteboul, S., Buneman, P., and Suciu, D. 2000. Data on the Web: From relations to semistructured data and XML. Data Management Systems. Morgan Kaufmann.

[11] Myllymaki, J. 2002. Effective Web data extraction with standard xml technologies. Computer Networks, 39 (5).

[12] Baumgartner, R., Flesca, S., and Gottlob, G. 2001. Visual Web information extraction with lixto. In VLDB Conference, (pp. 119–128).

[13] Crescenzi, V., Mecca, G., and Merialdo, P. 2001. RoadRunner: Towards automatic data extraction from large Web sites. In VLDB Conference, (pp. 109–118).

[14] Beneventano, D., Bergamaschi, S., Guerra, F., and Vincini, M. 2003. Synthesizing an integrated ontology. IEEE Internet Computing, 7 (5), (pp. 42–51).

[15] Castano S., De Antonellis V., De Capitani di Vimercati S, 2001. Global viewing of heterogeneous data sources. IEEE TKDE, 13(2).

[16] Klusch, M., Ossowski, S., and Shehory, O., 2002. Integrating Distributed In Sources with CARROTII. 6th International Workshop, CIA 2002.

[17] Bowman, C.M., Danzig, P., Hardy,D.R.,Manber, U., and Schwartz, M.F. 1995. Information discovery and access system. Computer Networks and ISD 28, 119–125.

[18] Callan, J.P., Lu, Z., and Croft, W.B. 1995. Searching distributed collections with networks. In Fox, E.A., Ingwersen, P., Fidel, R., eds.: SIGIR'95, Proceeding 18th Annual International ACM SIGIR Conference on Research and De in Information Retrieval. Seattle, Washington, USA.

[19] Woelk, D., and Tomlinson, C. 1995. Infosleuth: Networked exploitation of information semantic agents. In: COMPCON Conference.

[20] McClean, S. I., Karali, I., Scotney, B. W., Greer, K., Kapos, G. D., Hong, J., Bell, D. A., and Hatzopoulos, M. 2002. Agents for querying distributed statistical databases over the internet. Int. Journal On Artificial Intelligence Tool, 11, pp. 63-94.