Journal of Universal Computer Science, vol. 13, no. 12 (2007), 1936-1969
submitted: 15/4/07, accepted: 31/7/07, appeared: 1/12/07 © *J.UCS*

The SEWASIE Network of Mediator Agents for Semantic Search

Domenico Beneventano, Sonia Bergamaschi, Francesco Guerra
Maurizio Vincini
(University of Modena and Reggio Emilia, Italy
{domenico.beneventano, sonia.bergamaschi, francesco.guerra, maurizio.vincini}@unimore.it)

Abstract: Integration of heterogeneous information in the context of Internet becomes a key activity to enable a more organized and semantically meaningful access to data sources. As Internet can be viewed as a data-sharing network where sites are data sources, the challenge is twofold. Firstly, sources present information according to their particular view of the matter, i.e. each of them assumes a specific ontology. Then, data sources are usually isolated, i.e. they do not share any topological information concerning the content or the structure of other sources. The classical approach to solve these issues is provided by mediator systems which aim at creating a unified virtual view of the underlying data sources in order to hide the heterogeneity of data and give users a transparent access to the integrated information. In this paper we propose to use a multi-agent architecture to build and manage a mediators network. While a single peer (i.e. a mediator agent) independently carries out data integration activities, it exchanges knowledge with other peers by means of specialized agents (i.e. brokers) which provide a coherent access plan to access information in the peer network. This defines two layers in the system: at local level, peers maintain an integrated view of local sources, while at network level agents maintain mappings among the different peers. The result is the definition of a new networked mediator system intended to operate in web economies, which we realized in the SEWASIE (SEmantic Webs and AgentS in Integrated Economies) project. SEWASIE is a RDT project supported by the 5th Framework IST program of the European Community successfully ended on September 2005.

Keywords: Distributed databases, Distributed applications, Query language, Internet, Ontologies, Mediator, Knowledge Management
Categories: C.2, C.2.4, H.2, H.2.3, M.0, M.1

1 Introduction

The capillary diffusion of the Internet has made available access to an overwhelming amount of data, allowing users having benefit to vast information. However, information is not really directly available: internet data are heterogeneous and spread over different places, with several duplications and inconsistencies. In these cases, a data searching operation becomes an expensive task, due to the data incoherence and inconsistence solving and managing processes. The integration of such heterogeneous 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 [Garcia-Molina, 97, Chawathe, 93, Kirk, 95, Bergamaschi, 01]. The research on mediator-based systems has produced several results, but some issues, concerning the scalability, the methodology for building the ontology and the optimization in data access are still open. These issues are emphasized in the hidden web [Raghavan, 01], where several data-intensive web sites have to be integrated to provide relevant information in a specific domain.

Moreover, the centralized architecture of a mediator system presents several limitations: 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.

Data and metadata may frequently change in the web and the system must take these evolutions into account. We think that the use of agents may improve the work of the system by making it autonomous and able to adapt its activity to the dynamics of the network. Thus, we developed a semi-automatic technique for managing the network dynamics which we describe in the paper.

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 introduces an overview of the querying process and section 5 describes the main SEWASIE agents. Finally, section 6 describes some related works and section 7 sketches out some conclusions and future works.

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

represents 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 [Weiss, 00]. 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 be accomplished 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.

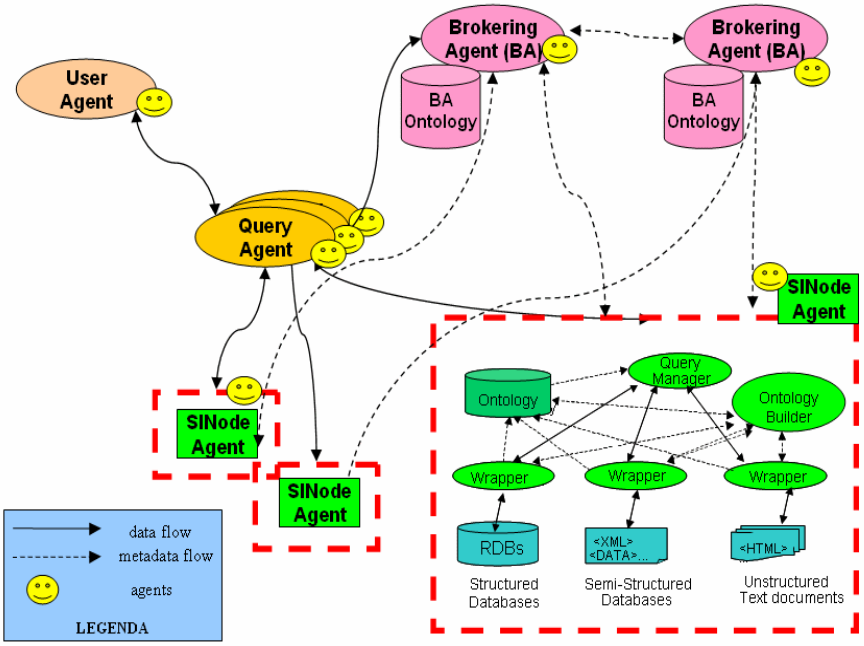


Figure 1: SEWASIE Network Architecture

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. 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.

The SEWASIE Information Nodes (SINodes) consist of a specific Ontology, created by an Ontology Builder, and a Query Manager, 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 information sources, wrappers, and a metadata

repository. The information sources managed by a SINode are heterogeneous collections of structured, semi-structured, or unstructured data, e.g. relational databases, XML/HTML or text documents. The interactions between the Ontology and the external data sources are handled by means of specific wrappers which are intended to translate to and from local access languages/structures.

The Ontology is synthesized by means of different steps which exploit the metadata of the sources and perform a semantic enrichment process by using the WordNet lexical database, artificial intelligence and clustering techniques. The methodology is supported by a tool, Ontology Builder, that will be described in detail in Section 6. The Ontology is exposed in OWL language and contains annotations w.r.t. WordNet and a set of mappings between the Ontology itself and the integrated sources that are exploited in the query processing phase, where a query may involve more than one SINode.

These mappings follow the Global-As-View (GAV) approach[1] [Lenzerini, 02], where the global view is obtained in term of the data sources.

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.

Therefore, Brokering Agents are responsible for maintaining a view of the knowledge handled by the network. This view is composed of:

- an Ontology that holds the information of the specific contents of the SINodes registered by itself;
- mappings between the Brokering Agent Ontology and the SINode Ontologies;
- mappings among the Brokering Agent Ontology and other brokering agent ontologies.

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 a Brokering Agent is contacted, it informs the Query Agent which SINodes under its control contain information relevant to query. Then, the query agent questions the relevant SINodes for collecting partial results. Also, it decides whether to continue the search with the other brokering agents. 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.

[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

3 SEWASIE Ontology creation

As stated in the previous section, 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 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 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.

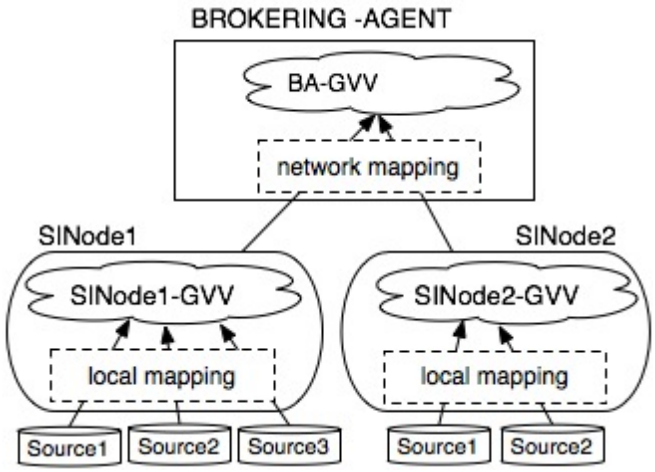


Figure 2: The BA/SINodes ontologies and mappings

The BA/SINodes architecture is realized via a two-level data integration system. From an architectural point of view, this architecture introduces a high level of flexibility as it permits integration between both data sources and data integration systems, which have already been developed independently (on the basis, for example, of sector or regional objectives). 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, [Madhavan, 03] 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

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

mappings, it was proved that it is possible to consider a "global" mapping as the composition of local and network mapping.

The two-level approach defines the whole Integration System as a triple IS = (GVV, N, M) constituted of:

- a GVV, which is a schema expressed in ODLI3 [Bergamaschi, 01] (a short description of the ODLI3 language is in APPENDIX B) and exposed in OWL language, composed mainly of global classes and global attributes;
- a set N of local sources: each local source has a schema also expressed in ODLI3;
- a set M of GAV mapping assertions between the GVV and N, where each assertion associates a global class C in the GVV to a query QN over the schemata of a set of local sources in N.

More precisely, for each global class C of the GVV, we define:

- a (possibly empty) set of local classes, denoted by L(C), belonging to the local sources in N;
- a conjunctive query QN over L(C);
- a lexical relationship with one or more terms in WordNet.

Intuitively, the GVV is the intensional representation of the information provided by the Integration System, whereas the mapping assertions specify how such an intensional representation should be related to the local sources managed by the Integration System. The semantics of an Integration System, and then of the SEWASIE system, is defined in [Calì, 04, Beneventano, 05].

Both SINode and BA Ontologies are defined as Integration Systems:

- an SINode is an Integration System SINode = (GVV, N, M) where the local sources N are data sources.
- a BA is an Integration System BA = (GVV, N, M) where the local sources N are SINodes, i.e., N = {$SINode_1$, $SINode_2$, . . . , $SINode_n$}.

3.1 Ontology Creation with MOMIS

The GVV classes and the mapping assertions (mappings for short) have to be defined during design by the Ontology Designer. This is done by using the Ontology Builder graphical interface, built upon the MOMIS framework. MOMIS (Mediator envirOnment for Multiple Information Sources) is a framework performing information extraction and integration from both structured and semi-structured data sources, plus a query management environment able to process incoming queries through the navigation of the mediated schema [Beneventano, 03].

The methodology for building the ontology of an SINode and of a BA is similar; we will first describe the methodology for an SINode and discuss later on the differences for a BA ontology.

The methodology consists of two main steps:

1. Ontology Generation

The system detects semantic similarities among the relevant source schemata, and automatically generates a GVV and the mappings between the GVV and the local schemata;

2. Mapping Refinement

The Ontology Designer interactively refines and completes the automatic integration results: in particular, the mappings automatically created by the system can be fine–tuned, and the query associated to each global class redefined by a set of functions that provide for data conversion and transformation.

3.1.1 An Example in the Mechanical Domain

We show the methodology applied on 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.

During the first integration level, two different SINodes-GVVs have been built by means of the MOMIS Ontology Builder: the first one includes the sites www.subfor.net and www.plasticitalia.com; the second SINode integrates the sites www.tuttostampi.com and www.deformazione.it.

A more complete description of the web sites is presented in Appendix A.

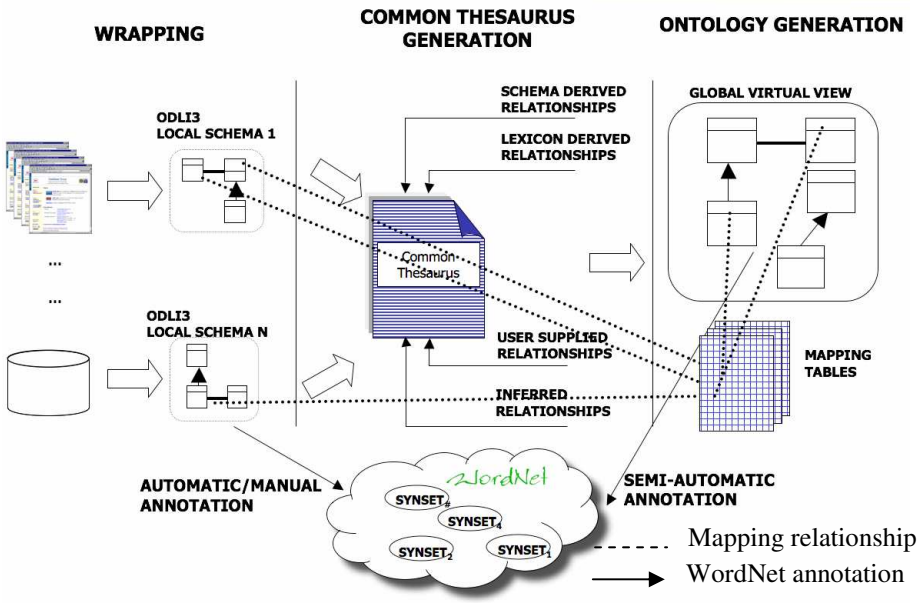


Figure 3: Ontology Generation Process for an SINode

3.1.2 GVV Generation Process

The Ontology Generation process can be outlined as follows (see Figure 3):

1. Extraction of Local Source Schemata: Wrappers acquire schemata of the involved local sources and convert them into ODLI3. 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 [Abiteboul, 00]. To perform information extraction and integration from HTML pages, research and commercial web data extraction tools, such as ANDES [Myllymaki, 02], Lixto [Baumgartner, 01] and RoadRunner [Crescenzi, 01], have been experimented and adopted to wrap the selected web sites.
   Figure 4 shows a portion of the XML-Schema of the www.plasticitalia.com site generated by Lixto and the translated ODLI3 representation.
2. 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%.
3. 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 (that are part of ODLI3). 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 adds new relationships to capture specific domain knowledge, and finally, by means of a Description Logics reasoner, ODB-Tools [Beneventano, 03], which performs equivalence and subsumption computation) infers new relationships and computes the transitive closure of Common Thesaurus relationship. In the example, including both SINode1 and SINode2, 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.

XML Schema

```
<xs:element name="company">
<xs:complexType> <xs:sequence minOccurs="0" maxOccurs="unbounded">
  <xs:element name="COMPANY_ID" type="xs:int" minOccurs="1"/>
  <xs:element name="NAME" type="xs:string" minOccurs="0"/>
  <xs:element name="ADDRESS" type="xs:string" minOccurs="0"/>
  <xs:element name="CITY" type="xs:string" minOccurs="0"/>
  <xs:element name="PHONE" type="xs:string" minOccurs="0"/>
  <xs:element name="FAX" type="xs:string" minOccurs="0"/>
  <xs:element name="EMAIL" type="xs:string" minOccurs="0"/>
  <xs:element name="WEB_SITE" type="xs:string" minOccurs="0"/>
  <xs:element name="INFO" type="xs:string" minOccurs="0"/>
 </xs:sequence> </xs:complexType> </xs:element>
<xs:element name="category">
<xs:complexType> <xs:sequence minOccurs="0" maxOccurs="unbounded">
  <xs:element name="CATEGORY_ID" type="xs:int" minOccurs="1"/>
  <xs:element name="NAME" type="xs:string" minOccurs="0"/>
</xs:sequence> </xs:complexType>
</xs:element>
<xs:element name="category_list">
<xs:complexType> <xs:sequence minOccurs="0" maxOccurs="unbounded">
  <xs:element name="CATEGORY_ID" type="xs:int" minOccurs="1"/>
  <xs:element name="COMPANY_ID" type="xs:int" minOccurs="0"/>
</xs:sequence> </xs:complexType>
</xs:element>
<xs:element name="plasticitalia">
<xs:complexType> <xs:choice maxOccurs="unbounded">
  <xs:element ref="company" />
  <xs:element ref="category" />
  <xs:element ref="category_list" />
</xs:choice></xs:complexType>
</xs:element> …
```

ODLI3

```
source XSD SN-MechNew
(description="plasticitalia" ) {
  interface category_list {
    attribute short  CATEGORY_ID;
    attribute short  COMPANY_ID; };
  interface  company {
    attribute short  COMPANY_ID;
    attribute string  NAME;
    attribute string  ADDRESS;
    attribute short  CITY;
    attribute string  PHONE;
    attribute string  FAX;
    attribute string  EMAIL;
    attribute string  WEB_SITE;
  };
  interface ${category} {
    attribute short CATEGORY_ID;
    attribute string  NAME;
  };
}
```

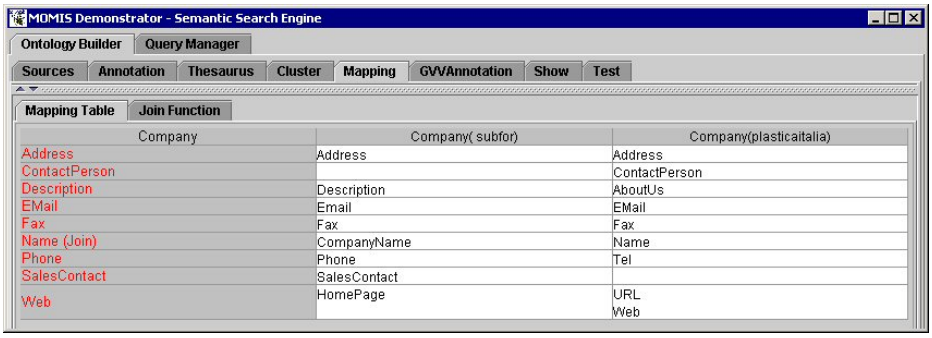
Figure 4: a portion of the XML-Schema and ODLI3 representation of www.plasticitalia.com

4. 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 clustered in the same global class by means of the ARTEMIS tool [Castano, 01]. ARTEMIS determines the degree of matching of two classes, based on their names and their structure, and produces an affinity tree. 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; in particular, the mappings which have been automatically created by the system can be fine-tuned as discussed in next section (Mapping Refinement). For example, the obtained SINode1's GVV is shown in figure 6.a, where the main global classes are Company, Country, Province, List_of_category, Category and a set of classes composing a (partial) hierarchy of the managed categories.
5. 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 annotation of a GVV is a significant result, since these metadata may be exploited in the BA Ontology building process, and the GVV annotation can be useful to make the meaning of the created domain ontology understandable to external users and applications [Beneventano, 03]. For both SINodes of the example each global class, the annotation is obtained automatically. As an example, we report the following global class meaning:

Global class	Local Classes	Meaning (from WordNet)
SINode1.Company	Subfor.Company, plasticitalia.company	Company#1
SINode1.Province	Subfor.Province	Province#1
SINode1.Category	Plasticitalia.category	Category#2

3.1.3 Mapping Refinement

The system automatically generates a Mapping Table (MT) for each global class C of the GVV, whose columns represent the local, classes L(C) belonging to C and whose rows represent the global attributes of C. An element MT [GA][LC] represents the set of local attributes of LC which are mapped onto the global attribute GA. Figure 5 shows part of the MT of the global class "Company" that collects the local classes Company of www.subfor.net and Company of www.plasticaitalia.com.



Company	Company(subfor)	Company(plasticaitalia)
Address	Address	Address
ContactPerson		ContactPerson
Description	Description	AboutUs
EMail	Email	EMail
Fax	Fax	Fax
Name (Join)	CompanyName	Name
Phone	Phone	Tel
SalesContact	SalesContact	
Web	HomePage	URL Web

Figure 5: Mapping Table of Company

The query QN associated to a global class C is implicitly defined by the Ontology Designer starting from the MT of C. The Ontology Designer can extend the MT by adding:

- Data Conversion Functions from local to global attributes
- Join Conditions between pairs of local classes belonging to C
- Resolution Functions for global attributes to solve data conflicts of local attribute values.

According to the resulting MT, the system automatically generates a query QN associated to C, by extending the Full Disjunction operator [Galindo-Legaria, 94], been recognized as providing a natural semantics for data merging queries [Rajaraman, 96].

Data Conversion Functions

The Ontology Designer can define, for each not null element MT[GA][L], a Data Conversion Function, denoted by MTF[GA][L], which represents the mapping of local attributes of L into the global attribute GA. MTF[GA][L] is a function that has to be executable/supported by the class L local source. For example, considering relational sources, MTF[GA][L] should be an SQL value expression. T(L) denotes the local class value L transformed by the Data Conversion Function.

As an example, in the graphic interface shown in figure 6, the designer could define MTF[Web][plasticaitalia.Company]: URL + Web where the symbol '+' stands for the standard SQL operator for string concatenation.

Join Conditions

Merging data from different sources requires different instantiations of the same real world object to be identified; this process is called object identification [Naumann, 02]. The topic of object identification is currently a very active research area with significant contributions both from the artificial intelligence [Tejada, 01] and database communities [Ananthakrishna, 02, Chaudhuri, 03]. We assume, for the sake of simplicity, that the ontology designer be able to define join conditions among local classes.

The Join Conditions among pairs of local classes belonging to the same global class are introduced in order to identify instances of the same object and fuse them.

Given two local classes L1 and L2 belonging to C, a Join Condition between L1 and L2, denoted with JC(L1,L2), is a Boolean expression of atomic constraints (L1.Ai Op L2.Aj) where Ai (Aj) are global attributes with a not null mapping in L1 (L2) and Op is a relational operator.
As an example, in figure 7, the designer can define:
JC(L1, L2) : L1.CompanyName = L2. Name
where L1= subfor.Company and L2= plasticaitalia.Company.

Resolution Functions
The fusion of data coming from different sources and taking into account the problem of inconsistent information among sources is a hot research topic [Di Giacomo, 04, Bertossi, 03, Greco, 03, Naumann, 02, Lin, 98]. In MOMIS the approach proposed in [Naumann, 02] has been adopted i.e. a Resolution Function for solving data conflicts may be defined for each global attribute mapping onto local attributes coming from more than one local source.

A global attribute with no data conflicts (i.e. the instances of the same real object in different local classes having the same value for this common attribute), is called Homogeneous Attribute. Of course, for homogeneous attributes, resolution functions are not necessary (a global attribute mapped onto only one source is a particular case of an homogeneous attribute).

As an example, in Company we defined all the global attributes as homogeneous attributes except for Address, where we used a precedence function:

L1.Company.Address has a higher precedence than L2.Company.Address.

Full Disjunction
QN is defined in such a way that it contains a single tuple resulting from the merge of all the different tuples representing the same real world object. This problem is related to that of computing the natural outer-join of many relations in a way that preserves all possible connections among facts ([Rajaraman, 96]). Such a computation has been termed as Full Disjunction (FD) by Galindo Legaria ([Galindo-Legaria, 94]).

In our context: given a global class C composed of L1, L2, ..., Ln, we consider FD(T (L1),T (L2),...,T (Ln)), computed on the basis of the Join Conditions.

The problem is how to compute FD. With two classes, FD corresponds to the full (outer) join: FD(T (L1),T (L2)) = T (L1) full join T (L2) on (JC(L1,L2)).

With more than 2 local classes, the computation of FD is performed as follows. We assume that: (1) each L contains a key, (2) all the join conditions are on key attributes, and (3) all the join attributes are mapped into the same set of global attribute, say K. Then, it can be proved that: (1) K is a key of C, and (2) FD can be computed by means of the following expression (called FDExpr):

```
(T(L1) full join T(L2) on JC(L1,L2))
  full join T(L3) on (JC(L1,L3) OR JC(L2,L3))
    ... full join T(Ln) on (JC(L1,Ln) OR JC(L2,Ln) OR ... OR JC(Ln-1,Ln))
```

Finally, QN is obtained by applying Resolution Functions to the attributes resulting from FDExpr: for a global attribute GA we apply the related Resolution Function to T (L1).GA, T (L2).GA, . . . , T (Lk).GA; this query QN is called FDQuery.

3.1.4 The Brokering Agent Ontology

Let us assume that the www.subfor.net and www.plasticaitalia.com sites have been integrated into SINode1, where the obtained GVV is shown in Figure 6.a, while the sites (www.tuttostampi.com and www.deformazione.it) have been integrated into another SINode (SINode2); a subset of the obtained GVV is shown in Figure 6.b. (light arrows indicate ISA relationships, while dark arrows PART-OF relationships).

When a GVV is completed, the SINode Agent proposes the list of available Brokering Agents on the SEWASIE network and the SINode manager decides to apply the GVV to one or more Brokering Agent: for example, in Figure 7 the SINode manager decides to contact for the subscription the MechanicBA.

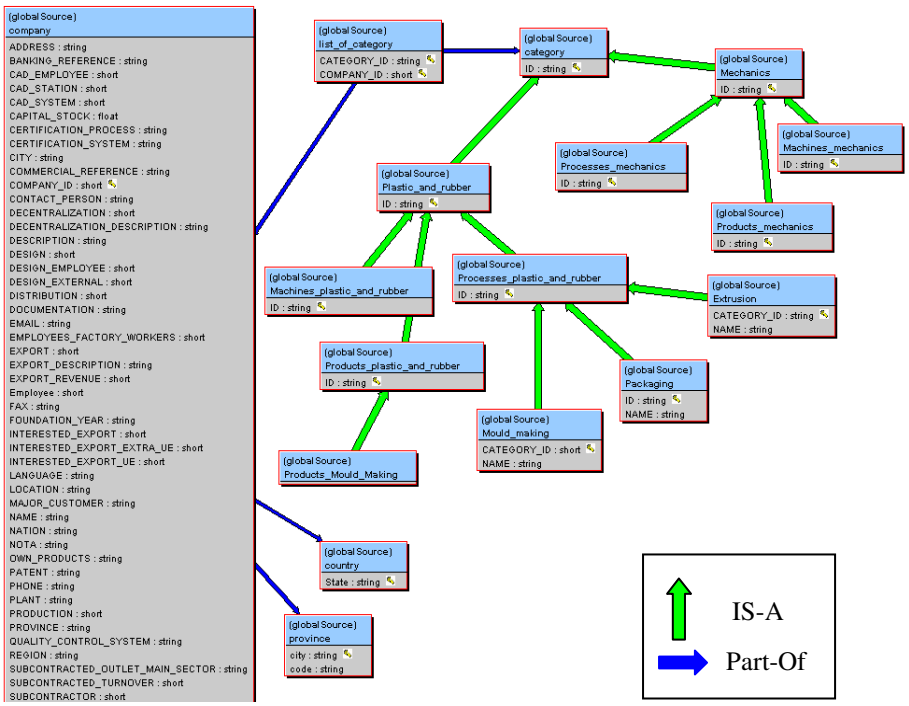
When one or more SINode Agent request the subscription to a BA, the BA manager can decide to agree or refuse the request (see Figure 7, where the MechanicBA accepts the subscription of SINode1 and SINode2). If the BA agrees, an acknowledge message is sent to the relevant SINodes and the MOMIS system initiates the process of creation/update of the BA-GVV (or its updating).

The BA-GVV generation process is performed starting from step 3, i.e. extraction and annotation are not necessary, as SINode-GVVs are annotated ODLI3 schemata.

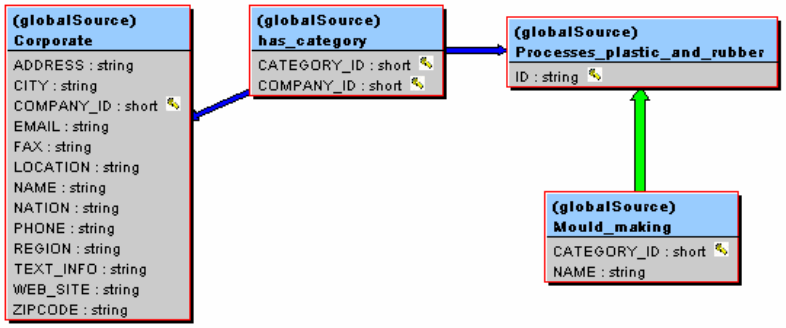
With references to our example, the two SINodes of figure 6 have been associated to the same BA (MechanicBA), as they refer to the same domain ontology. The following semantic relationships:

1. *Company SYN Corporate*
2. *list_of_category SYN has_category*
3. *Mould_making NT Processes_plastic_and_rubber*
4. *Processes_plastic_and_rubber NT Processes_plastic*

are discovered by the system on the basis of the meanings of the above elements in the annotated SINode-GVVs. The automatically built BA Ontology (figure 8) contains the following main global classes: Company (which includes Company of SINode1 and Corporate of SINode2), list_of_category (has_category and list_of_category), Category, Country, Province. The system recalculates the Category hierarchy, in particular the hyponymy between Process_plastic of SINode1 and Process_plastic_and_rubber of SINode2 and, and the category_list of SINode2 is automatically referred to the whole Category hierarchy.



(a)



(b)

Figure 6: (a) SINode1 GVV *(b) SINode2* GVV

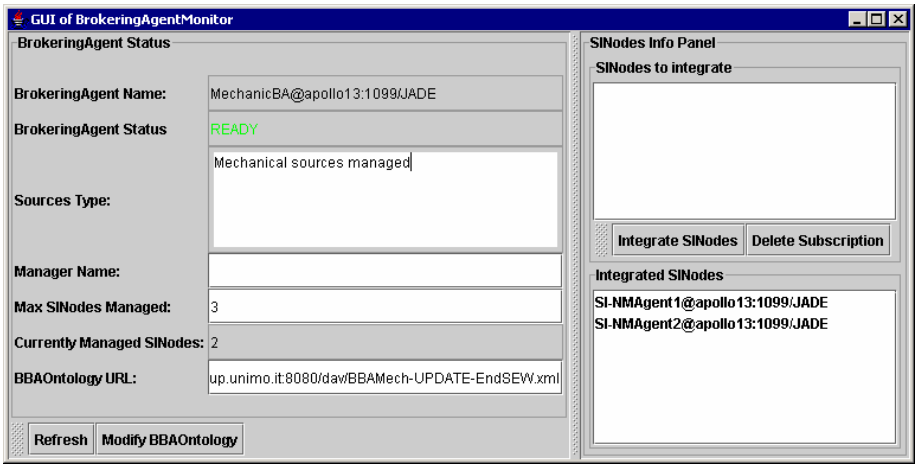


Figure 7: MechanicBA agent accepts the SINodes subscriptions

3.2 Adding New Sources to an SINode

In the ontology research area, many solution to the challenge of supporting an ontology's evolution have been proposed. In Section 6 we briefly present the two major approaches: the evolution approach tries to face the problem of dynamics in its whole complexity, while the versioning approach distinguishes between different versions of ontologies to reduce the complexity. We propose the use of a single ontology that will auto-arrange to maintain the consistence with the source that it represents.

An SINode must modify the ontology whenever new sources are added or deleted on the network or when existing sources change, and these modifications must be exported to Brokering Agents and Query Agents.

Since the integration process is expensive for both the designer and the system, we propose a methodology that reuses the results of the initial GVV, rather then restarting the integration process from scratch. The ontology building approach assumes that all the sources to be integrated contribute with the same weight to the process. Therefore, if we assume that the world described by an SINode is quite static and referencing the same specific context and the new source belongs to this context, we may assume that a new source brings less semantics than the previously built GVV. For this reason, we may assume an integration process of the new source that starts from the obtained GVV and its lexicon annotation, and tries to integrate this new source in the GVV.

Our approach to the process consists of managing the integration of two schemata, which is quite general and can be applied both to an SINode and a BA Ontology.

The system treats the GVV's global classes as local classes and integrates them with the new source's local classes. The process introduces the following notation:

- gcNew is a global class of the new integrated schema. It has a name (gcNewName) and a set of global attributes ($gcNewAtt_i$).

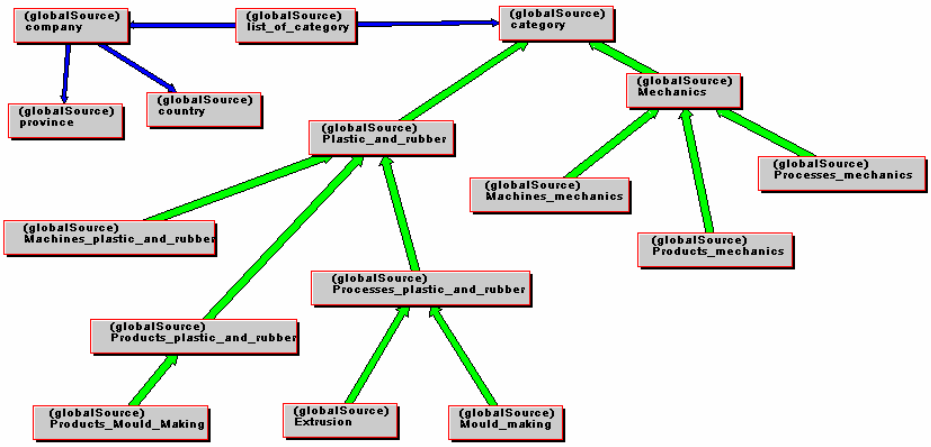


Figure 8: The Brokering Agent Ontology

- gcOld is a global class of the old integrated schema. It has a name (gcOldName) and a set of global attributes (gcOldAtt$_j$).
- lcNew is the local class of the new source. It has a name (lcNewName) and a set of local attributes (lcNewAtt$_k$).

According to the Momis integration methodology, we must create a common thesaurus. In this case, the common thesaurus contains schema-derived relationships extracted from the new source and intra-schema lexicon-derived relationships obtained from annotating the new source. Furthermore, we have to semantically enrich the GVV global classes using this semiautomatic annotation method. Interestingly, the GVV annotation lets us discover inter-schema lexical relationships, thus enriching the common thesaurus.

Let us suppose that the simple new SINodes should be added to the BA Ontology: each class is automatically annotated and these annotations are use to enrich the Common Thesaurus.

The next step is cluster generation, which is followed by the creation of the global classes and mapping tables. This phase provides mapping rules between GCs and new or old local classes. By integrating the old integrated schema and the new source we obtain a new integrated schema whose global classes gcNew comprise both old global classes gcOld$_l$ and new local classes lcNew$_j$:

$$gcNew = \{gcOld_1,...,gcOld_p,lcNew_1, ...,lcNew_n\}$$

A new global class gcNew is expressed as a set of local classes (new or old) by substituting gcOld$_i$ with the respective old local classes lcOld$_{ik}$.

$$gcNew = \{lcOld_{11},...,lcOld_{1z},..., lcOld_{p1},..., lcOld_{pn},lcNew_1,..., lcNew_n\}$$

Figure 9: New SINode for BA ontology

With global class generation, we observe that, using the same clustering parameters, an old global class, $lc_1,...,lc_i,...,lc_n$, changes only if the integration process inserts one or more new local classes ($lcNew_i$) into the global class. We therefore find two different interesting cases.

Scenario 1
A new global class gcNew is composed at least of one old global class gcOld and one or more new local classes $lcNew_i$. For example,

$gcNew = \{gcOld, lcNew_1,...,lcNew_i,..., lcNew_n\}$

The gcNew might have new GAs generated from the new local classes' semantic contributions. Ontology Builder defines new mapping rules between a global attribute and its corresponding local attributes. In this case, GAs belonging to gcOld ($gcOldAtt_i$) can map both local classes of the old global class and new local classes. New global attributes can map only new local classes (null mappings). The meaning of old GAs must be enriched with the meanings of the new local classes mapped by these attributes, and the meaning of new global attributes must be set according to the rules defined in the global attributes annotation.

In the example, the Enterprise class of the new SINode is automatically included in the BBA Company class and the Enterprise's attributes are encapsulated in the Company's attributes. For example:
Enterprise.NAME Company.Name
Enterprise.WEB_SITE Company.web
Enterprise.FAX Company.FaxNumber

Scenario 2
A global class of the new integrated schema is composed only of new local classes.

$gcNew = \{lcNew_1,...,lcNew_i,...,lcNew_n\}$

In this situation, the GVV is extended without interfering with the previous one. As stated, the gcNew has a name (gcNewName) and a set of new global attributes ($gcNewAtt_i$), each mapping only new local attributes. The names and meanings of the

global attributes are defined following the rules stated in the global attributes annotation.

In our example, the packaging class gives rise to a new global class, while the Processes_and_plastic class is included in Process_plastic_and_rubber existent class, and the hierarchy is maintained thanks to the lexical hyponym relationship present in WordNet between Processes plastic and Packaging. The new BA Ontology automatically obtained is shown in Figure 10.

4 Querying the SEWASIE System

A complete description of the query processing with respect to the two-level ontology architecture is included in [Beneventano, 06b]. In this section, this methodology is extended according to the Multi-Agent System infrastructure. This section is organized as follows. After a brief description of the query reformulation process for the two-level data integration system, full outer join optimization techniques and the agent-based prototype for query processing implemented in the SEWASIE system are described.

4.1 Query Formulation

The User Agent, accessible by means of a generic HTML browser (an online version is available at http://www.sewasie.org/sewasie-prototype.htm), shows the BA Ontology in terms of classes and properties and allows the user to pose a query in a graphical manner. A conjunctive query composed of unary (classes) and binary (attribute and associations) terms is automatically created by the User Agent starting from the query posed by the user exploiting the graphical interface.

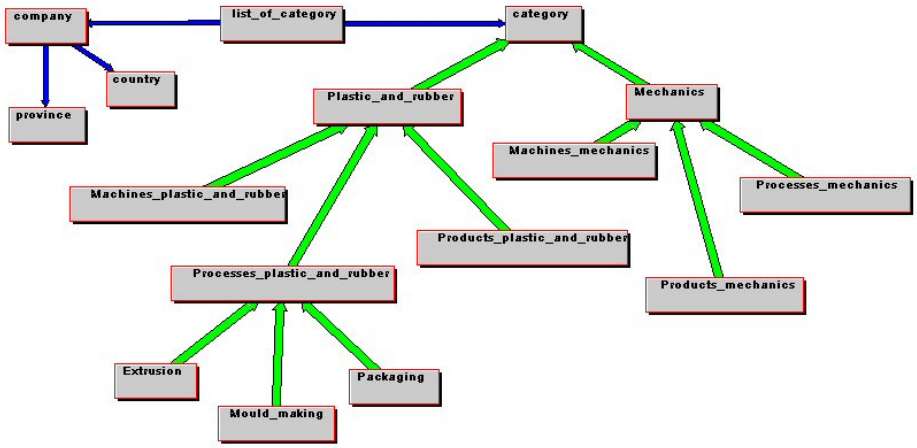


Figure 10: Final BA ontology after the insertion of a new SINode

For example, the user may graphically compose the following query:

Name and Address of the companies located in the Veneto Italian Region with a capital stock greater than 50 with plastic and rubber products

The query is automatically translated by the system into the following conjunctive query:

Q = { (X1,X2) | Company(X), Name(X,X1), Address(X,X2),
Region(X,X3), EqualTo(X3,'Veneto'),
Capital_Stock(X,X4), GreaterThan(X4,50),
List_of_Category(X,X5),
Products_Plastic_and_Rubber(X5)
}

where

- *Company*, *List_of_Category* and *Products_Plastic_and_Rubber* are classes
- *Name*, *Address*, *Region* and *List_of_Category* are attributes
- *GreaterThan* and *EqualTo* are standard comparison predicates of the BA ontology.

4.2 Query Reformulation

Query reformulation takes into account the two different levels of mappings, giving rise to two different steps for query answering:
1) Reformulation w.r.t. the BA Ontology and 2) Reformulation w.r.t. the SINode Ontology.
These two reformulation steps are similar and they include the processes of:

1. Query expansion: the query posed in terms of the ontology (GVV) is expanded to take into account the explicit and implicit constraints: all the constraints in the GVV are compiled in the expansion, so that the expanded query can be processed by ignoring constraints. Then, the atoms (i.e. subqueries referring to a single global class) are extracted from the expanded query.
2. Query unfolding: the atoms in the expanded query are unfolded by exploiting the mappings M between the GVV and the local sources in N.

In the following section, we show an example of query expansion (the algorithm for Query expansion is reported in [Di Giacomo, 04]) and we discuss the unfolding process of an atom starting from the query QN associated to a global class.

4.2.1 Query Expansion Example

The output of the query expansion process are an expanded query (called EXPQuery) and its atoms (called EXPAtoms); EXPQuery is a union of conjunctive queries on the GVV; an EXPAtom is a Single Class Query on a Global Class of the GVV.
As an example, the Query Expansion process for the previous query Q, produces:

- EXPQuery = Q1 $\vee$ Q2

where Q1 = Q and

Q2 = { (X1,X2) | Company(X), Name(X,X1), Address(X,X2),
Region(X,X3), EqualTo(X3,'Veneto'),
Capital_Stock(X,X4), GreaterThan(X4,50),

List_of_Category(X,X5), Products_Mould_Making (X5)
}

i.e., Q2 takes into account the constraint *Products_Mould_Making ISA Products_Plastic_and_Rubber.*

• A set of EXPAtoms:

ExpAtom1 = { (X1,X2) | Company(X), Name(X,X1), Address(X,X2), Region(X,X4), EqualTo(X4,'Veneto'), Capital_Stock(X,X5), GreaterThan(X5,50)}
ExpAtom2 = { X6 | Company(X), List_of_Category(X,X6) }
ExpAtom3 = { X6 | Products_Plastic_and_Rubber (X6)}
ExpAtom4 = { X6 | *Products_Mould_Making* (X6)}

4.2.2 Query Unfolding

The query unfolding process is performed for each EXPAtom which is a query Q over a global class C of the GVV (for the 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).

5 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 ([TILab, 00]). In this section we introduce the main tasks performed by the SEWASIE agents.

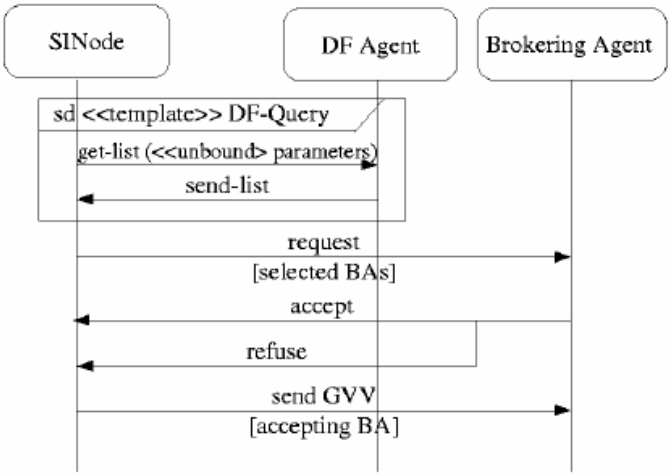


Figure 11: The interaction between an SINode agent and a Brokering agent

As described in the previous section, the first step towards integrating data must be undertaken at the level of SINodes, where the Ontology Builder creates and maintains a global virtual view of its information sources. After that, an SINode Agent must ask to be included in a BA of the network (the list of available BAs is provided by the Directory Facilitator agent offered by the JADE platform).

Figure 11 shows the AUML sequence diagram describing the creation of the Brokering Agent GVV in SEWASIE. After the SINode request, the BA manager may accept this new source, and consequently the BA GVV is updated. On the other hand, the BA may decide to refuse the new SINode, by notifying a message to the SINode Agent. In order to manage this process, we developed a communication protocol which is able to describe the different status of the agents. In particular, in order to avoid inconsistencies, it is possible for a user to query the BA only if its status is READY (see Figure 12), i.e. its GVV can represent all of the registered SINodes. The SINode agent may assume five different status levels, as shown in Figure 12; among them, the most representative levels are: GVV CREATED, a GVV is built including all the managed data sources; STAND-BY, the SINode has requested to be included into a BA GVV and it is waiting for this registration; READY, the SINode is almost included into a BA GVV.

5.1 User Agents

UAs are in charge of all search activities in the SEWASIE network, therefore they mainly have a coordination role. They are created whenever the Query Interface is instantiated, and their first task is to find their BA of reference. Once the user defines the query, these agents have to translate the query into the internal language, pass it to a new Query Agent, and wait for the results. 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.

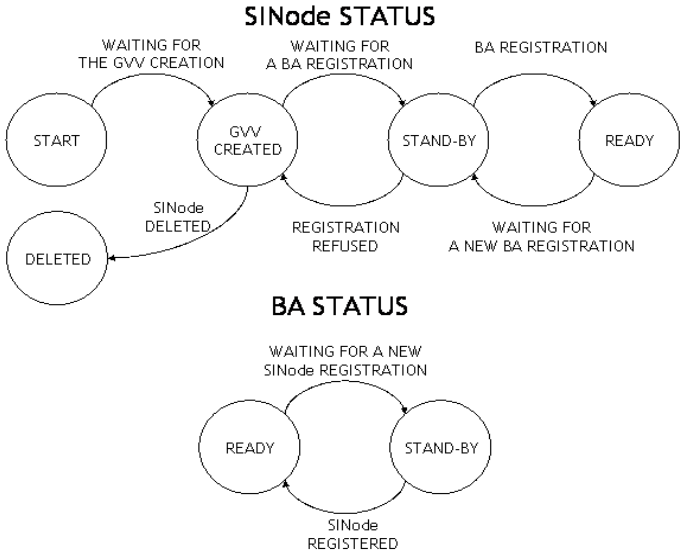


Figure 12: The different status level for the Brokering and SINode Agents

5.2 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 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 its only implemented service (solve-query), 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.

5.3 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.

5.4 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. 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.

6 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 [Klusch, 02] 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 [Bowman, 95], CORI [Callan, 95] and InfoSleuth [Woelk, 95]. Other systems, like TSIMMIS [Chawathe, 94], include some rewriting rules against predefined query patterns. There are several steps of query processing also in the MISSION project [McClean, 02]. In these cases, data integration technology is not present or, as in TSIMMIS, is limited to automatic generation of wrappers [Hammer, 97] and mediators [Papakonstantinou, 95] from web pages. In SEWASIE, the data integration techniques [Beneventano, 03] adopted by SINodes address unstructured and semi-structured data sources, as well as relational databases.

In the area of distributed data-sharing, the Peer Data Management System (PDMS) are the natural extension to distributed databases within the context of P2P system [Herschel , 05]. A PDMS is a decentralized architecture for web-scale data sharing: peers are autonomous concerning the data they store locally and the semantic description and conceptual organization they provide for the data they want to share with other peers.

In the Piazza PDMS [Halevy, 03] every peer is associated with a schema that represents the peer domain of interest, and semantic relationships between peers are provided locally between pairs (or small sets) of peers in the form of mappings.

The Humboldt Discover [Herschel, 05] enables a peer to locate information sources in a PDMS which is not reachable by schema mapping by exploiting Semantic Web technologies to abstract from the concrete data model of the peer and to index the peer schema.

A promising approach to semantic self-organization of autonomous communities of peers is proposed in HELIOS [Castano, 05], where semantic affinity function is defined on each peer pairwise: the semantic affinity, like in MOMIS approach, is based on the language (derived from WordNet) and the context.

The SEWASIE Network should be considered a PDMS where each peer (SINode) is provided with an ontology that represents the peer domain of interest and peers are managed by the BAs, i.e. super-peers that connect SINodes through local semantic mappings.

Several surveys about the approaches in the ontology management area have been published. This topic is generally divided into three categories: ontology development, ontology and schema matching and ontology alignment.

Concerning the ontology development, the ONTOWEB project published a complete technical report (http://www.ontoweb.org deliverable 1.4) where tools are classified on the basis of the implemented methodologies (from scratch, reengineering ontologies, based on a cooperative construction, and managing the evolution). Several researchers address topics in the ontology matching area, i.e. the techniques for identifying similar concepts in different ontologies: in [Rahm, 01] several systems are evaluated on the basis of the generated mappings (five kinds of criteria are identified), while [Noy, 04] focuses on mapping discovery, reasoning and representation. The ontology alignment, i.e. the automated resolution of semantic correspondences between the elements of heterogeneous ontologies, is one of the new challenge in the ontology management and it includes ontology mapping and schema mapping. The Knowledgeweb Network of Excellence (knowledgeweb.semanticweb.org) has largely investigated about this issue publishing several reports.

In the following paragraphs, we shall compare some ontology management system with our approach.

Clio [Miller, 01] is a research prototype providing to the user a graphical interface in order to support the creation of mappings between two data representations. There are many differences between Clio and MOMIS: first, in the Clio framework the focus is on schema mapping issues, while the focus of our proposal is the semi-automatic generation of a "target" schema common to each source (the Global Virtual View). Moreover, our proposal relies on structural and lexical relationships between the sources.

COMA [Do, 02] (and COMA++ [Aumueller, 05]) is a composite matcher which provides an extensible library of different matchers and supports various aggregating and selecting strategies. Matchers exploit linguistic, data-type, and structural information, as well as previous matches, to produce similarity matrices. Particular similarity values are then selected as suitable match candidates, and combined into a single value. This process is performed for whole schemata or for two schema elements, and is repeated after the user provides feedback. COMA supports a reuse approach focusing on existing mappings, which can be generalized for different reuse granularities, or fragment- and schema-level match results. The starting mappings (or similarity) are user-defined, unlike our approach that is mainly focused on the use of lexical dictionaries (like WordNet) to discover semantic relationships .

GLUE and iMAP [Doan, 02] are an extension of LSD system [Doan, 01] whose goal is to semi-automatically find schema mappings for data integration. Like its

ancestor LSD, Glue uses machine learning techniques to find mappings [Doan, 04]. It first applies statistical analysis to the available data (joint probability distribution computation). It then generates a similarity matrix, based on the probability distributions, for the data considered and uses "constraint relaxation" in order to obtain an alignment from the similarity, which is obtained by using probabilistic definition of several similarity measures. This approach relies on data instances techniques. On the other hand, the MOMIS methodology is based on schema analysis (we are experimenting the introduction of instance based components, see [Beneventano, 06c] for some preliminary results).

FCA–MERGE [Stumme, 01] follows a bottom-up approach, which offers a global structural description, for merging ontologies. For the source ontologies, it extracts instances from a given set of domain-specific text documents by applying natural language processing techniques. Based on the extracted instances, a mathematical technique taken from Formal Concept Analysis [Ganter, 04] is applied to derive a lattice of concepts as a structural result of FCA–MERGE. The generated result is then explored and transformed into the merged ontology with human interaction. As with our approach, FCA-MERGE uses a linguistic techniques to generate a context for the ontology, and uses a shallow text processor for German [Neumann, 97] and not a lexicon database such as WordNet.

PROMPT [Noy, 00] is an algorithm that provides a semi-automatic approach to ontology merging and alignment embedded in Protege 2000. It starts with the identification of matching class names. Based on this initial step, an iterative approach is carried out for performing automatic updates, finding resulting conflicts, and offering suggestions to remove these conflicts. The initial list of matches based on class names can be computed by any linguistic concept similarity algorithm defined by other system, and uses Protege component-based architecture to allow the user to plug in any term-matching algorithm, so that, in theory, our approach can be applied in conjunction with PROMPT algorithm.

Researchers have proposed many solutions to the challenge of supporting an ontology's evolution, but two major methods have emerged. The ontology evolution [Motik, 02] approach employs the timely adaptation of an ontology as well as the consistent propagation of changes. A modification in one part of the ontology can create subtle inconsistencies in other parts of the same ontology, dependent ontologies, and applications. When a change occurs, it is vital to ensure the consistency of the ontology and all dependent artefacts. The Karlsruhe Ontology and Semantic Web framework (Kaon), for example, is based on this approach (http://kaon.semanticweb.org). The other leading approach, ontology versioning, can be defined as the ability to handle changes in ontologies by creating and managing different variants [Klein, 01]. Such methodologies must be able to distinguish and recognize versions, and include procedures for updating and changing ontologies. This approach is used with the Simple HTML Ontology Extensions (Shoe; www.cs.umd.edu/projects/plus/SHOE/), a small extension to HTML that lets Web page authors annotate their documents with machine-readable knowledge.

7 Conclusion and Future Work

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.

Acknowledgements

This work was supported in part by the 5th Framework IST programme of the European Community through project SEWASIE within the Semantic Web Action Line. The SEWASIE consortium comprised in addition to the authors' organization (Sonia Bergamaschi was the coordinator of the project), the Universities of Aachen RWTH (M. Jarke), Roma La Sapienza (M. Lenzerini, T. Catarci), Bolzano (E. Franconi), as well as IBM Italia (G. Vetere), Thinking Networks AG (C. Engels) and CNA (A. Tavernari) as user organisation.

References

[Abiteboul, 00] Abiteboul, S., Buneman, P., Suciu, D.: Data on the Web: From relations to semistructured data and XML. Data Management Systems. Morgan Kaufmann, 2000.

[Ananthakrishna, 02] Ananthakrishna, R., Chaudhuri, S., Ganti, V.: Eliminating fuzzy duplicates in data warehouses. In VLDB Conference, (pp. 586–597), 2002.

[Aumueller, 05] Aumueller, D., Do, H. H., Massmann, S., Rahm, E.: Schema and ontology matching with COMA++. SIGMOD Conference, (pp. 906-908), 2005.

[Baumgartner, 01] Baumgartner, R., Flesca, S., Gottlob, G.: Visual Web information extraction with lixto. In VLDB Conference, (pp. 119–128), 2001.

[Beneventano, 03] Beneventano, D., Bergamaschi, S., Guerra F., Vincini, M.: Synthesizing an integrated ontology. IEEE Internet Computing, 7 (5), (pp. 42–51). 2003.

[Beneventano, 05] Beneventano, D., Lenzerini, M.: Final release of the system prototype for query management. Sewasie, Deliverable D.3.5, www.dbgroup.unimo.it/prototipo/paper/D3.5 final.pdf, 2005.

[Beneventano, 06a] Beneventano, D., Bergamaschi, S.: Semantic search engines based on data integration systems. In Semantic Web Services: Theory, Tools and Applications. Idea Group., 2006.

[Beneventano, 06b] Beneventano, D., Bergamaschi, S., Nana Mbinkeu, C.R.: Full Outer Join Optimization Techniques in Integration Information Systems. Sewasie, DII Technical Report, www.dbgroup.unimo.it/prototipo/paper/DIITR_Beneventano_2006.pdf, 2006.

[Beneventano, 06c] Beneventano, D., Bergamaschi, S., Bruschi, S., Guerra, F., Orsini, M., Vincini, M.: Instances Navigation for Querying Integrated Data from Web-Sites. In WEBIST Conference, (pp 46-53), 2006.

[Bergamaschi, 01] Bergamaschi, S., Castano, S., Beneventano, D., Vincini, M.: Retrieving grating data from multiple sources: the MOMIS approach. Data Knowl. Eng. 36, (pp. 215–249), 2001.

[Bertossi , 01] Bertossi, L. E., Chomicki, J.: Query answering in inconsistent databases. In J. Chomicki, R. van der Meyden, & G. Saake (Eds.), Logics for Emerging Applications of Databases (pp. 43–83). Springer, 2001.

[Bowman, 95] Bowman, C. M., Danzig, P., Hardy, D. R., Manber, U., Schwartz, M. F.: Information discovery and access system. Computer Networks and ISD 28, (pp. 119–125), 1995.

[Calì, 04] Calì, A., Calvanese, D., Di Giacomo, G. D., Lenzerini, M.: Data integration under integrity constraints. Inf. Syst., 29 (2), (pp. 147–163), 2004.

[Callan, 95] Callan, J. P., Lu Z., Croft, W. B.: Searching distributed collections with networks. In Fox, E.A., Ingwersen, P., Fidel, R., Proceeding 18th Annual International ACM SIGIR Conference on Research and De in Information Retrieval. Seattle, Washington, USA, July 9-13, (pp 21–28), 1995.

[Castano, 01] Castano, S., De Antonellis, V., De Capitani di Vimercati, S.: Global viewing of heterogeneous data sources. IEEE Transactions on Data and Knowledge Engineering, 13(2), 2001.

[Castano, 05] Castano S., Montanelli, S.: Semantic Self-Formation of Communities of Peers. In Proc. of the ESWC Workshop on Ontologies in Peer-to-Peer Communities, Heraklion, Greece, 2005.

[Chaudhuri, 03] Chaudhuri, S., Ganjam, K., Ganti, V., Motwani, R.: Robust and effcient fuzzy match for online data cleaning. In ACM SIGMOD Conference, (pp. 313–324), 2003.

[Chawathe , 94] Chawathe, S., Garcia-Molina, H., Hammer, J., Ireland, K., Papakonstantinou, Y., Ullman, J. D., Widom, J.: The TSIMMIS project: Integration of heterogeneous information sources. In: 16th Meeting of the Information Processing Society of Japan, Tokyo, Japan, 7–18, 1994.

[Crescenzi, 01] Crescenzi, V., Mecca, G., Merialdo, P.: RoadRunner: Towards automatic data extraction from large Web sites. In VLDB Conference, (pp. 109–118), 2001.

[Di Giacomo, 04] Di Giacomo, G. D., Lembo, D., Lenzerini, M., Rosati, R.: Tackling inconsistencies in data integration through source preferences. In ACM IQIS Workshop (pp. 27–34), 2004.

[Do 02] Do, H. H., Rahm, E.: COMA - A System for Flexible Combination of Schema Matching Approaches. . In VLDB Conference, (pp. 610-621), 2002.

[Doan, 01] Doan, A. H., Domingos, P., Halevy, A. Y.: Reconciling Schemas of Disparate Data Sources: A Machine-Learning Approach. In SIGMOD Conference, (pp.509-520), 2001.

[Doan, 02] Doan, A. H., Madhavan, J., Domingos, P., Halevy, A. Y.: Learning to map between ontologies on the semantic web. In WWW Conference, (pp. 662-673), 2002.

[Doan, 04] Doan, A. H., Madhavan, J., Domingos, P., Halevy, A. Y.: Ontology Matching: A Machine Learning Approach. Handbook on Ontologies, (pp 385-404), 2004.

[Galindo-Legaria, 94] Galindo-Legaria, C. A.: Outerjoins as disjunctions. In ACM-SIGMOD Conference, (pp. 348–358), 1994.

[Galindo-Legaria, 97] Galindo-Legaria, C. A., Rosenthal, A.: Outerjoin simplification and reordering for query optimization. ACM Trans. Database Syst., 22(1), (pp 43–73), 1997.

[Garcia-Molina, 97] Garcia-Molina, H., Papakonstantinou, Y., Quass, D., Rajaraman, A., Sagiv, Y., Ullman, J. D., Vassalos, V., Widom, J.: The TSIMMIS Approach to Mediation: Data Models and Languages. J. Intell. Inf. Syst. 8(2), (pp 117-132), 1997.

[Ganter, 04] Ganter, B., Stumme, G., Wille, R.: Formal Concept Analysis: Theory and Applications. J. UCS 10(8), 2004.

[Greco, 03] Greco, G., Greco, S., Zumpano, E. : A logical framework for querying and repairing inconsistent databases. IEEE Trans. Knowl. Data Eng., 15 (6), (pp 1389–1408), 2003.

[Halevy, 03] Halevy, A. Y., Ives, Z. G., Suciu, D., Tatarinov, I.: Schema mediation in peer data management systems. In Proc. of the 19th ICDE Conference, 2003.

[Hammer, 97] Hammer, J., McHugh, J., Garcia-Molina, H., Semistructured data: The Tsimmis experience. In ADBIS97 Conference, (pp. 1-8), 1997.

[Herschel , 05] Herscheland, S., Heese, R.: Humboldt Discoverer: A Semantic P2P index for PDMS. In Proc. of the International Workshop Data Integration and the Semantic Web, Porto, Portugal, 2005.

[Kirk, 97] Kirk, T., Levy, A. Y., Sagiv, Y., Srivastava, D.: The Information Manifold. In Knoblock, C., Levy, A. eds, Stanford, 1995.

[Klein, 01] Klein, M., Fensel, D.: Ontology Versioning on the Semantic Web, First Int. Semantic Web Working Symp., Stanford University Press, (pp. 75–91), 2001.

[Klusch, 02] Klusch, M., Ossowski, S., Shehory, O.: Integrating Distributed In Sources with CARROTII. 6th International Workshop CIA, Madrid, September 18-20, 2002.

[Larson, 05] Larson, P. A., Zhou, J.: View Matching for Outer-Join Views. In VLDB Conference, (pp 445-456), 2005.

[Lenzerini, 02] Lenzerini, M.: Data Integration: A Theoretical Perspective. In PODS Conference, (pp 233-246), 2002.

[Lin, 98] Lin, J., Mendelzon, A. O.: Merging databases under constraints. Int. J. Cooperative Inf. Syst., 7 (1), (pp 55–76), 1998.

[Madhavan, 03] Madhavan, J., Halevy, A. Y.: Composing Mappings Among Data Sources. In VLDB Conference, (pp. 572-583), 2003.

[McClean, 02] McClean, S. I., Karali, I. , Scotney, B. W., Greer, K., Kapos, G. D., Hong, J., Bell, D. A., Hatzopoulos, M.: Agents for querying distributed statistical databases over the internet. Int. Journal On Artificial Intelligence Tool, 11, (pp. 63-94), 2002.

[Miller, 01] R. J. Miller, Hernández, M. A., Haas, L. M., Yan, L., Ho, C. T. H., Fagin, R., Popa, L.: The Clio Project: Managing Heterogeneity. SIGMOD Record 30(1), (pp 78-83), 2001.

[Motik, 02]B. Motik et al.: User-Driven Ontology Evolution Management, Proc. 13th Int'l Conf. Knowledge Eng.and Knowledge Management, LNCS 2473, Springer, (pp. 285–300), 2002.

[Myllymaki, 02] Myllymaki, J.: Effective Web data extraction with standard xml technologies. Computer Networks, 39 (5), (pp 635–644), 2002.

[Naumann, 02] Naumann, F., Haussler, M.: Declarative data merging with conflict resolution. In MIT-IQ Conference, (pp. 212–224), 2002.

[Neumann, 97] Neumann, G., Backofen, R., Baur, J., Becker, M., Braun, C.: An Information Extraction Core System for Real World German Text Processing. In ANLP Conference, (pp 209-216), 1997.

[Noy, 00] Noy, N. F., Musen, M. A.: PROMPT: Algorithm and Tool for Automated Ontology Merging and Alignment. In AAAI/IAAI Conference, (pp 450-455), 2000.

[Noy, 04] Noy, N. F.: Semantic Integration: A Survey Of Ontology-Based Approaches. SIGMOD Record 33(4), (pp 65-70), 2004.

[Papakonstantinou, 95] Papakonstantinou, Y., Garcia-Molina, H., Widom, J.: Object Exchange across heterogeneous information source. In ICDE Conference, (pp. 251-260), 1995.

[Raghavan, 01] Raghavan, S., Garcia-Molina, H.: Crawling the Hidden Web. In VLDB Conference, (pp 129-138), 2001.

[Rahm, 01] Rahm, E., Bernstein, P. A.: A survey of approaches to automatic schema matching. VLDB Journal 10(4), (pp 334-350), 2001.

[Rajaraman, 96] Rajaraman, A., Ullman, J. D.: Integrating information by outerjoins and full disjunctions. In ACM-PODS Conference (pp. 238–248), 1996.

[Rao, 04] Rao, J., Pirahesh, H., Zuzarte, C.: Canonical abstraction for outerjoin optimization. In SIGMOD Conference, 2004.

[Stumme, 01] Stumme, G., Maedche, A.: FCA-MERGE: Bottom-Up Merging of Ontologies. In IJCAI Conference, (pp 225-234), 2001.

[Tejada, 01] Tejada, S., Knoblock, C. A., Minton, S.: Learning object identification rules for information integration. Inf. Syst., 26 (8), (pp 607–633), 2001.

[TILab, 00] TILab, T.I.L.: JADE. jade.cselt.it, 2000.

[Weiss, 00] Weiss, G.: Multigent Systems – A Modern Approach to Distributed Artificial Intelligence. Cambridge, MA: MIT Press, (pp. 79-120), 2000.

[Woelk, 95] Woelk, D., Tomlinson, C.: Infosleuth: Networked exploitation of information semantic agents. In COMPCON Conference, 1995.

APPENDIX A

Web URL	www.subfor.net
Type	"Yellow pages" site.
Subject	Subcontracting, instrument for rapidly finding suppliers who can meet specific production needs.
Languages	Italian, English and German.
Database size	Contains information about 5240 companies from 8 Italian regions and 9 business activities.
Data	Information about companies that work in the sectors: Plastic and rubber, Mechanics, Wood, Textiles, Electronics, Knitwear, Tanning, Footwear Peltry. For each company (described in detail), several services and products are indicated (about 1500 categories of services and goods were wrapped).
Home Page	

Web URL	www.plasticaitalia.com
Type	"Yellow pages" site.
Subject	"Yellow pages" for the Italian plastic companies. Companies pay to be added to the database.
Languages	Italian and English
Database size	The database contains information about 6500 companies. These companies are categorized on the basis of 467 categories of services and goods.
Data	It is possible to select a company by means of a three level hierarchy. The first level contains 6 classes, the second one specializes that classes into 60 subclasses.
Home Page	

Web URL	www.tuttostampi.com
Type	"Yellow pages" site.
Subject	The mission of Tuttostampi is to help the industrial molding companies make their way into the new economy world. To such purpose they provide training, information and service support, and a space for the new kinds of commercial transactions.
Languages	Italian and English.
Database size	4000 Italian companies.
Data	Companies (described by company name, address, city, area/district, zip code, country, phone and fax number, email, products, web site, free text description) are catalogued by means of 22 different categories.
Home Page	

Web URL	www.deformazione.it
Type	"Yellow pages" site.
Subject	Portal for Italian companies working on metallic sheets.
Languages	Italian, English (only for company sectors).
Database size	The database contains 2244 companies that are represented by means of 39 categories.
Home Page	

APPENDIX B

The *ODLI3* language

ODLI3 is an extension of the object-oriented language ODL (Object Definition Language) (http://www.odmg.org) which is a specification language used to define the interfaces to object types that conforms to the ODMG object model introduced for information extraction. ODLI3 extends ODL with constructors, rules and relationships useful in the integration process both to handle the heterogeneity of the sources and to represent the GVV. In particular, ODLI3 extends ODL with the following relationships that express intra- and inter-schema knowledge for source schemata:

- Synonym of (SYN) relationships are defined between two terms ti and tj that are synonyms;
- Broader terms (BT) relationships are defined between two terms ti and tj, where ti has a broader, more general meaning than tj. BT relationships are not symmetric.
- Narrower terms (NT) relationships are the opposite of BT relationships.
- Related terms (RT) relationships are defined between two terms ti and tj that are generally used together in the same context in the considered sources.

ODLI3 also extends ODL with the addition of integrity-constraint rules, which declaratively express if-then rules at both the intra- and inter-source level. ODLI3 descriptions are translated into the Description Logic OLCD - Object Language with Complements allowing Descriptive cycles - (Beneventano, Bergamaschi, Lodi, and Sartori, 1988), in order to perform inferences that will be useful for semantic integration.

Because the ontology is composed of concepts (represented as global classes in ODLI3) and simple binary relationships, translating ODLI3 into a Semantic Web standard such as RDF, DAML+OIL, or OWL is a straightforward process. In fact, from a general perspective, an ODLI3 concept corresponds to a class of the Semantic Web standards, and ODLI3 attributes are translated into properties. In particular, the IS-A ODLI3 relationships are equivalent to subclass-of in the considered Semantic Web standards. Analyzing the syntax and semantics of each standard, further specific correspondences might be established. For example, there is a correlation between ODLI3's simple domain attributes and the DAML+OIL DataTypeProperty concept. Complex domain attributes further correspond to the DAML+OIL ObjectProperty concept (www.w3.org/TR/daml+oil-reference).