SYNTHESIZING AN INTEGRATED ONTOLOGY WITH MOMIS

Roberta Benassi, Domenico Beneventano, Sonia Bergamaschi, Francesco Guerra and Maurizio Vincini
Dipartimento di Ingegneria dell'Informazione - Università di Modena e Reggio Emilia
Via Vignolese 905 - Modena
{lastname.firstname}@unimore.it

Abstract – The Mediator EnvirOnment for Multiple Information Sources (MOMIS) aims at constructing synthesized, integrated descriptions of the information coming from multiple heterogeneous sources, in order to provide the user with a global virtual view of the sources independent from their location and the level of heterogeneity of their data. Such a global virtual view is a conceptualization of the underlying domain and then may be thought of as an ontology describing the involved sources. In this article we explore the framework's main elements and discuss how the output of the integration process can be exploited to create a conceptualization of the underlying domain

Keywords: Heterogeneous Sources, Mediator, Global as view, WordNet, Knowledge Representation and Ontologies, Internet and WWW

I. INTRODUCTION

To exploit the Internet's expanding data collection, current Semantic Web approaches employ annotation techniques to link individual information resources with machine-comprehensible metadata. Before we can realize the potential this new vision presents, however, there are several issues that must be solved. One of these is the need for data reliability in a dynamic, constantly changing network. Another issue is how to explicitly specify relationships between abstract data concepts. Ontologies provide a key mechanism for solving these challenges, but the dynamic nature of the web leaves the question of how to manage them.

The Mediator Environment for Multiple Information Sources (MOMIS) aims at constructing synthesized, integrated descriptions of the information coming from multiple heterogeneous sources, in order to provide the user with a global virtual view of the sources independent from their location and the level of heterogeneity of their data. Such a Global Virtual View (GVV) is a conceptualization of the underlying domain and then may be thought of as an ontology describing the involved sources. The Semantic Web exploits semantic markups referring to ontologies' items to provide web pages with meanings easily and completely understandable by machines. Therefore, the Semantic Web is based on a "a priori" existence of ontologies representing the specific domain of the sources. This approach relies on the accuracy of the selected reference ontology; our assessment is that the most commonly used ontologies are generic and then the annotation phase, i.e. the phase where semantic annotations connecting web page parts to ontology items are provided, causes loss of semantics. By means of the involved sources, our approach builds an ontology that exactly represents the domain. Moreover, the GVV is annotated according to a lexical ontology, providing in this way an easily-understandable meaning to its content. In this article, we use Web documents as a representative information source to describe the MOMIS methodology's general application. We explore the framework's main elements and discuss how the output of the integration process can be exploited to create a conceptualization of the underlying domain. This paper, which is an extension of [1] is organized as follows: Section II introduces the MOMIS's framework and provides a description of the ODL $_{I3}$ language as a common data model for integrating a given set of local information sources. Section III explains how MOMIS builds a domain ontology from scratch, while Section IV describes the GVV generation process, i.e. the process for creating a conceptualization of the integrated information sources. Section V concludes the paper by summarizing current developments and future work.

II. THE MOMIS'S FRAMEWORK

MOMIS is a framework for extracting information and integrating heterogeneous, semistructured information sources such as Web data sources (www.dbgroup.unimo.it/momis/). Unlike other data-integration systems that follow the local-as-view (LAV) approach, which is based on the idea that each source's content should be represented by predefined global schema, MOMIS implements a semiautomatic methodology that follows the global-as-view (GAV) approach[3]**:** the obtained global schema is expressed in terms of the data sources**.** More precisely, to each element of the global schema, a view over the data sources is associated, so that its meaning is expressed as the data residing at the sources. MOMIS uses ODL$_{I3}$, which is based on the Object Definition Language (ODL) to describe both the input (the sources) and the result of the synthesis process (global virtual view).

MOMIS generates a global schema that provides an integrated GVV composed of a set of global classes that represent the information contained in the underlying sources and the mappings that establish the connections among the global attributes of the global classes and the source schemata. Since a GVV conceptualizes a domain, it might be thought of as an ontology for the integrated sources.

A. The ODL$_{I3}$ language

ODL$_{I3}$ is an extension of the Object Definition Language (http://www.service-architecture.com/database/articles/odmg_3_0.html), which is used to define interfaces to object types that conform to the Object Data Management Group (ODMG) object model. ODL $_{I3}$ extends ODL with constructors, rules, and relationships that are useful in the ontology-integration process – both for handling source heterogeneity and representing the global virtual view (GVV). In particular, ODL $_{I3}$ extends ODL with several relationships that express intra and inter-schema knowledge for source schemas [2].

- `Synonym of (SYN)` relationships are defined between two terms t_i and t_j that share meanings.
- `Broader terms(BT)` relationships are defined between two terms t_i and $t_{j,}$ where t_i has a more general meaning than $t_{j.}$ BT relationships are not symmetric.
- `Narrower terms(NT)` relationships are the opposite of BT relationships.
- `Related terms(RT)` relationships are defined between two terms t_i and t_j that are generally used together in the same context in the considered sources.

ODL$_{I3}$ also extends ODL by adding *integrity-constraint rules*, which declaratively express `if-then` rules at both the intra- and intersource level. ODL$_{I3}$ descriptions are translated into the Object Language with Complements allowing Descriptive cycles (OLCD)[15][16] in order to perform inferences that will be useful for semantic integration.

Because the ontology is composed of concepts (represented as global classes in ODL$_{I3}$) and simple binary relationships, translating ODL$_{I3}$ into a Semantic Web standard such as RDF, DAML+OIL, or OWL is a straightforward process. In fact, from a general perspective, an ODL $_{I3}$ relationships translate into properties. In particular, the `is-a` ODL $_{I3}$ relationships are equivalent to `subclassof` in the considered Semantic Web standards. We might recognize further specific correspondences by analysing the syntax and semantics of each standard. For example, there is a correspondence between the ODL $_{I3}$ interface and DAML+OIL `class`. There is also a correlation between ODL $_{I3}$'s simple domain attributes and the DAML+OIL `DataTypeProperty` concept. Complex domain attributes further correspond to the DAML+OIL `ObjectProperty` concept (http://www.w3.org/TR/daml+oil-reference). Moreover, classes are wrapped in both approaches.

III. Building an Ontology

A. Local Source Schemata extraction

The first step when building an ontology with the MOMIS framework is to construct a semantic representation, or conceptual schema, of the information source using ODL$_{I3}$.We assign each source a wrapper that logically converts the underlying data structure, if there is, into the ODL$_{I3}$ information model. For this reason, the wrapper architecture and interface are crucial because wrappers are the focal point for managing the data sources' diversity. For conventional structured information sources (for example, relational and object-oriented databases), a schema description is always available and can be directly translated. For semistructured information sources (for example, web pages and XML documents, a schema description is not directly available. In fact, a basic characteristic of semistructured data is that they are "self-describing" —that is, the information associated with the schema is specified in the data. To manage a semistructured source, a specific wrapper has to implement an automatic methodology to extract and explicitly represent the source's conceptual schema.

In the MOMIS framework, we developed a wrapper in order to translate XML and document type definition (DTD) files, into ODL$_{I3}$ format. To manage information in HTML format, which does not separate data structure from layout, we needed another step of extraction via a HTML/XML wrapper. More precisely, HTML/XML wrappers are specialized programs that identify the data of interest in a Web page and map them to a more suitable format (e.g., XML), enabling their further automatic processing. Wrappers can be manually coded or generated by the so-called *web data extraction tools.* While the former approach is time-consuming and error-prone, the latter provides sophisticated toolkits to simplify and speed up the whole wrapper generation process. We tested several research and commercial tools (including RoadRunner[7], Andes[8] and Lixto[6]) under several point of view (e.g. degree of automation, quality of the data extraction process, ease of use and so on). We selected Lixto as the most suitable for our approach because il provides a fully visual and interactive interface that assists the user in semiautomatically creating a wrapper program. The ODL$_{I3}$ description shown in Figure 3 is acquired by means of a Lixto-generated HTML/XML wrapper (DTDs in Figure 2).

B. Local Source Annotation with WordNet

There are mainly two ideas behind exploiting a lexical ontology in MOMIS.

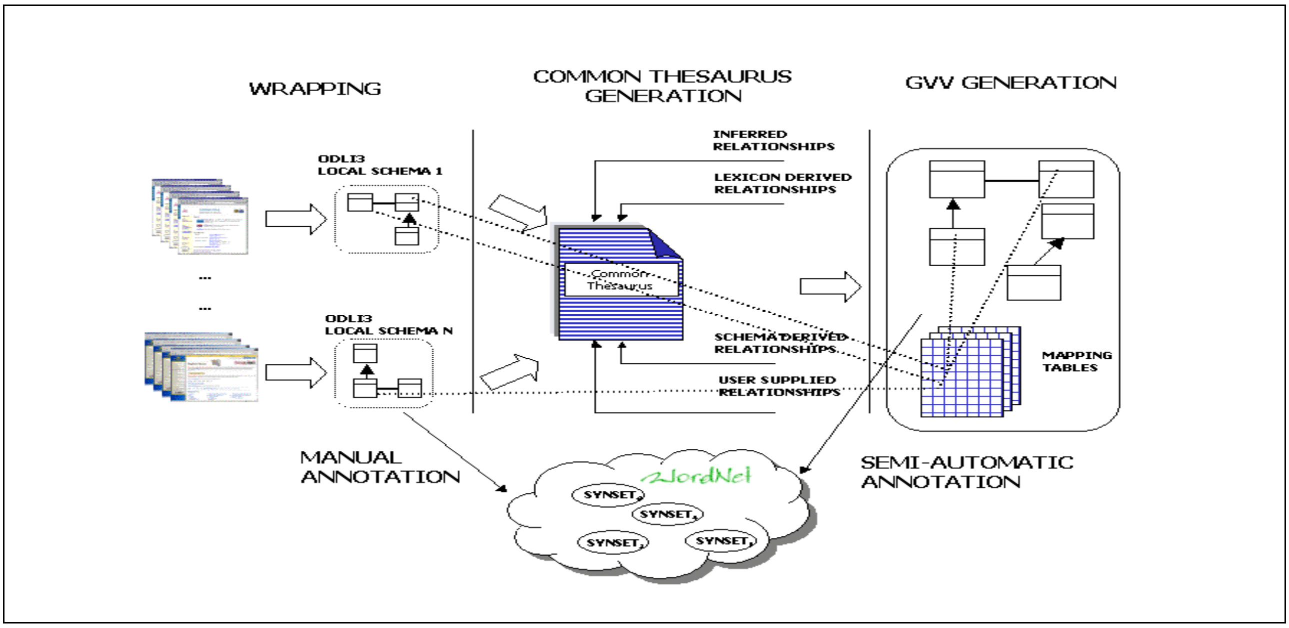

Figure 1. Overview of the ontology-generation process. The figure shows the local schemas' generation, where local schemas are annotated according to the lexical ontology WordNet, the Common Thesaurus generation, and finally the GVV global classes. In particular, these ones are connected by means of mapping tables to the local schemas and are (semi-automatically) annotated according to WordNet.

The first idea concerns the integration process: we think that terms used to describe schemas or structures in information sources hold exploitable semantics. The second idea concerns the use of the results of the integration process as domain ontology: in order to allow external users and applications to use our ontology, each its item has to have a well-known meaning. We chose the WordNet database as lexical ontology.

The WordNet database contains 146,350 lemma, organized in 111,223 synonym sets. WordNet's starting point for lexical semantics comes from a conventional association between the forms of the words — the way in which they are pronounced or written — and the concept, or meaning, they express. The association between the word's form and its meaning is synthesized in lexical matrix M (shown in Table I), in which rows show word meanings (each row represents a *synset*, i.e. a synonym set; a set of words that are interchangeable in some context) and columns show word forms (basic form or lemma).

Entry E1,1 implies that word form F1 can express word meaning M1. If there are at least two entries in the same column, the corresponding word form is *polysemous* — that is, it can be used to represent more than one meaning, (exactly two in this case); if there are at least two entries in the same row, the two word forms are synonymous.

Given a word form F, its *i*-th meaning will be denoted by F#*i*. For example, the word form "course" has 8 meanings in WordNet; the first is `course#1 = "education imparted in a series of lessons or class meetings"`.

Table I. Wordnet word forms and meanings.

	F1	**F2**	**F3**	**...**	**Fn**
M1	E1,1	E1,2			
M2		E2,2			
M3			E3,3		
...				...	
Mm					Em,n

C. Common Thesaurus Generation

The common thesaurus is constructed through a process that incrementally adds four types of relationships: schema-derived relationships, lexicon derived relationships, designer-supplied relationships and inferred relationships.

University Site (UNI)	Computer Science Site (CS)

`<!ELEMENT UNI(People*)>`
`<!ELEMENT People(Research_Staff*, School_Member*)>`
`...`
`<!ELEMENT Research_Staff(name, e-mail, Section*, Article*)>`
`<!ELEMENT Section(name, year. period)>`
`<!ELEMENT Article(title, year, journal, conference)>`
`<!ELEMENT School_Member(name, e-mail)>`
`<!ELEMENT name (#pcdata)> ...`
	`<!ELEMENT CS(Person*)>`
	`...`
	`<!ELEMENT Person(Professor*, Student*)>`
	`<!ELEMENT Professor(first_name, last_name, e-mail, Publication*)>`
	`<!ELEMENT Student(name, e-mail)>`
	`<!ELEMENT Course(denomination, Professor)>`
	`<!ELEMENT Publication(title, year, journal, editor)>`
	`<!ELEMENT School_Member(name, e-mail)>`
	`<!ELEMENT name (#pcdata)>...`

Figure 2. Document type definition (DTD) fragments used to represent source schemas. MOMIS uses these DTDs built with a Lixto-generated HTML/XML wrapper to translate source content for Web pages from a university and a computer science department into XML files.

1) ***Schema-derived relationships***. MOMIS automatically extracts these relationships, which are express at the intra-schema level, then schema-derived relationships are express at the intra-schema level, i.e., schema-derived relationships are express between elements of the same schema/, by analyzing each schema separately. For example, when analyzing XML data files, MOMIS generates BT and NT relationships from couples IDs and IDREFs (in an XML file an ID is an identifier for an element and an IDREF is a reference to an ID) and RT relationships from nested elements. Further extraction rules can be applied to other data models. For example, we extract intra-schema RT relationships from foreign keys in relational source schemas. In the relational model, a foreign key is a set of attributes of a relation used to express a reference from a relation to another. When a foreign key is also a primary key, in both the original and referenced relation, MOMIS extracts BT and NT relationships, which are derived from inheritance relationships in object-oriented schemas.

2) ***Lexicon-derived relationships.*** These originate from the annotation of the schemas respect the lexical ontology. WordNet defines a large variety of semantic relations between its meanings. A lexicon relationship between terms for the common thesaurus is derived from a semantic relation in WordNet between the meanings annotated for the terms according to the following correspondences:

- *synonymy* (similar relation) in WordNet corresponds to a SYN relationship in ODL_{I3};
- *hypernymy* (super-name relation) in WordNet corresponds to a BT relationship in ODL_{I3};
- *hyponymy* (sub-name relation) in WordNet corresponds to a NT relationship in ODL_{I3};
- *holonomy* (whole-name relation) in WordNet corresponds to a RT relationship in ODL_{I3};
- *meronymy* (part-name relation) in WordNet corresponds to a RT relationship in ODL_{I3}; and
- correlation (two term having the same hypernym) in WordNet corresponds to a RT relationship in ODL_{I3}.

Unknown terms do not give lexicon-derived relationships to the common thesaurus. Moreover, an incorrect annotation with respect to WordNet may generate wrong relationship inserted in the common thesaurus.

3) ***Designer-supplied relationships.*** To capture specific domain knowledge, designers can supply new relationships directly. This operation is crucial because the new relationships are forced to belong to the common thesaurus. If a meaningless or incorrect relationship is inserted, the subsequent integration process can produce a wrong global schema.

4) ***Inferred relationships***. MOMIS exploits description logic techniques from ODB-Tools[9] to infer new relationships by applying subsumption computation to "virtual schemas" obtained by interpreting BT and NT as subclass relationships and RT as domain attributes. A class C1 subsumes a class C2 if the description of C2 implies the description of C1; subsumption computation is performed, in our context, by a syntactically comparison of class descriptions. For example, if FEMALE NT PERSON, then the class C1 with description {attribute children PERSON } subsumes the class C2 with description {attribute children FEMALE}.

In the example described in Figures 2 and 3, for instance, MOMIS automatically obtained and proposed of the following relationships:

`CS.Professor NT CS.Person` [schema-derived]
`UNI.School_Member NT CS.Person` [lexicon-derived]
`UNI.Article NT CS.Publication` [lexicon-derived]

University Site (UNI)

```
                    …
Interface Research_Staff
(Source Un_site.dtd)
{ attribute string name;
  attribute string email;
  attribute set <Section> section;
  attribute set <Article> article;
}
Interface Article
(Source Un_site.dtd)
{ attribute string title;
  attribute string journal;
  attribute string conference;
  attribute string year;
}
...
```

Computer Science Site (CS)

```
                    …
Interface Professor
(Source Sc_site.dtd)
{ attribute string first_name;
  attribute string last_name;
  attribute string email;
  attribute set <Pubblication> publication;
}
Interface Publication
(Source Sc_site.dtd)
{ attribute string title;
  attribute string year;
  attribute string journal;
}
...
```

Figure 3. Pieces of the Univesity (UNI) and computer science (CS) sources in ODL$_{I3}$. MOMIS uses the XML/DTD wrapper to translate the generated DTDs into ODL $_{I3}$ descriptions.

```
UNI.Research_Staff NT CS.Person [inferred]
UNI.Research_Staff RT UNI.Article [inferred]
```

If the designer accepts and confirms these relationships, they are included in the common thesaurus.

D. Extending WordNet

Lexical semantic ontologies, such as WordNet, have proven very useful with many applications in Natural Language Applications. However, they usually only include general terms, as it would be impossible to extend them with every concept used in every domain of knowledge. In this context, we find very specific terms pertaining to different domains. If a source description element (i.e., a class name) does not find a correspondent within the reference lexical ontology (WordNet in our case), then the designer is requested to adapt the element to an already existing concept or to completely ignore it. However both these choices cause loss of information. In order to fully exploit semantics held in local schemata and improve semiautomatic annotation of the GVV, we developed a tool named WNEditor which makes the designer able to efficiently extend WordNet, by creating/managing new meanings and setting relationships between new meanings and pre-existing ones. Since WordNet is distributed *as-it-is*, external applications, such as MOMIS, are not allowed to directly modify its data files. Thus, we extrapolated the WordNet internal organization and a relational DBMS is employed to store original data and the added ones. The distinction between the former and the latter is achieved by introducing additional information such as the extension's name and its owner.

The WNEditor's philosophy is based on the awareness that the designer knows the organization in synsets of the WordNet lexical ontology. Despite this, the extension process is rather critical due to the hugeness and the complexity of the lexicon ontology. WNEditor helps the designer to perform step-by-step operations, e.g., creating a new synset and providing its definition (gloss), as follows:

1) ***Creating a new synset starting from an existing word form:*** the word form journal has in WordNet 5 meanings, with journal#2: a periodical dedicated to a particular subject; "he reads the medical journals", as the most appropriate. On the contrary, let us suppose that the designer does not find is satisfactory enough because the definition is too generic and lacks any references to scientific research work. In this case the designer can define a new meaning for the word form journal#NEW1: a periodical made of selected papers describing academic/industrial research work about a particular subject. Furthermore, the designer can eventually add other word forms pertaining to this new synset, for example, "scientific journal".

2) ***Creating a new synset starting from a new word form:*** when the word form and the proper meaning are not in the lexical database, the solution is to introduce both the word form and a new synset. As an example, let us suppose that the designer wants to introduce University_Member with the new meaning *"*one of the person who compose the university staff (especially professors and researchers*)".* Since meanings do not exist in isolation but are related each others, the designer has to add relationships between new synsets and those already existing. . In this way, all the new inserted elements (synsets, word form and relationships) are fully integrated in WordNet and can be used during other annotation phases.

Every WordNet relation (hypernymy/hyponymy, meronomy/holonomy, etc…) consists of two members, a source synset and a target synset. Therefore, given the new meaning `journal#NEW1: a periodical made of selected papers describing academic/industrial research work about a particular subject`, as the source synset, the designer is guided in searching for appropriate target synsets. Under the assumption that similar enough natural language definitions should also provide some evidence of concept similarity, we can obtain the target candidate synsets by exploiting an heuristics known in literature as definition match [14] and applying it to the WordNet's glosses. In essence, WNEditor automatically retrieves a list of candidate synsets sharing somewhat similarities with the source one. Then, the designer is asked to explicitly declare the type of lexical and semantic relationships (such as hypernymy/hyponymy, meronomy/holonymy and so on) to relate source synset to targets ones, if any.

IV. GLOBAL VIRTUAL VIEW GENERATION

The GVV consists of a set of Global Classes; for each Global Class, a Mapping Table is defined to connect the Global Attributes of the Global Class with the Local Attributes of the source schemas. To build the Global Classes MOMIS has to identify ODL_{I3} classes that describe the same or semantically related concepts in different sources. Therefore, the system defines *affinity coefficients*[10] for all possible pairs of ODL_{I3} classes, based on their relationships in the common thesaurus. Affinity coefficients determine the degree of matching between two classes, based on their names (*name affinity* coefficient) and attributes (*structural affinity* coefficient). MOMIS then calculates the linear combination of the two to create the *global affinity* coefficient, which a hierarchical clustering algorithm[11] then uses to classify ODL_{I3} classes. The clustering procedure output is an affinity tree, in which ODL_{I3} classes are the leaves and intermediate nodes have associated affinity values. MOMIS interactively computes the integration clusters from the affinity tree using a threshold-based mechanism for which the integration designer sets the parameters.

The generation of global classes from selected clusters is a synthesis activity that MOMIS performs interactively with the designer. It builds a global class GC_i definition for each cluster Cl_i. Once the Global Classes are constructed, MOMIS creates for each Global Class the corresponding Global Attributes' set.

This phase requires the interaction with the integration designer and consists of two phases. First, the system automatically associates a set of global attributes with GC_i, corresponding to the union of local attributes for the classes belonging to Cl_i. Using the common thesaurus lattice that contains SYN relationships and BT and NT relationships among local attributes, the system then proposes restrictions the designer might impose on the global attribute set.

For each global class, a persistent mapping table MT (like the one in Table II) stores all the generated mappings. First column in the table represent the global attributes of the select global class, the other ones represent the local classes belonging to the global class; rows represent the global attributes. An element MT[GA][LC] represents the set of attributes of the local class (LC) that are mapped to the global attribute (GA). The GA attribute value is a “mapping” function of the values assumed by the set of attributes MT[GA][LC]. Some simple and frequent cases of this mapping function are:

- *Identity*. The GA value is equal to the local attribute LA value; we denote this case as MT[GA][LC] = LA.
- *Conjunction*. The GA value is obtained as a conjunction of the values assumed by a set of local attributes LA_i for the local class LC; we denote this case as MT[GA][LC] = LA_1 and ... and LA_n.
- *Constant*. The GA assumes into the local class LC a constant value set by the designer; we denote this case as MT[GA][L] = const.
- *Undefined*. The GA is undefined for the local class LC; we denote this case as MT[GA][L] = null.

In our university Web page example, the integration process gives rise to three global classes:

```
Global1: (UNI.Section, CS.Course)
Global2: (UNI.Article, CS.Publication)
Global3: (UNI.Research_Staff,
UNI.School_Member, CS.Professor, CS.Student)
```

We report, as an example, the Mapping Table for Global2:

Table II. Mapping Table of the global class Global2 (Publication)

	UNI.Article	CS.Publication
Title	Title	Title
Year	Year	Year
Journal	Journal	Journal
Conference	Conference	null
Editor	Null	Editor

For more information on the process of generating GVVs, see the MOMIS project homepage (http://www.dbgroup.unimo.it).

A. Global Virtual View Annotation

GVV annotation assigns a global element name (GEN) and a set of global element meanings (GEM_i; a class or attribute meaning given by the disjunction of its set of meanings), to each global element (GE; class or attribute):

$$GE = \langle GEN, \{GEM1, \ldots , GEMp \}\rangle, p=0$$

In our work with the MOMIS project, we have developed a semiautomatic methodology for annotating a GVV.

1) ***Global Class Annotation***. To semiautomatically associate an annotation to each global class, we consider the set of all its "broadest" (A class C is broader than a class C' if C BT' C or C' NT C) local classes, with respect to the relationships included in the common thesaurus; this set is denoted by BLCGC (Broadest Local Classes of GC) and is defined as follows:

$$BLC_{GC} = \{LC \in GC \mid \neg\exists y \in, (LC\ NT\ y) \lor (y\ BT\ LC)\}$$

For the meanings in Table III, the designer would use GC_B to annotate the global class (GC) by name choice and meaning choice.

2) ***Name choice.*** To identify each GC and its contents, the integration designer selects a name to serve as a label — particularly to identify the GC's role. The system suggests a list of possible names, but the designer can also choose one that is not in the list. Therefore, a name might not be a WordNet word form. As shown in Table III, the designer selected the name `course` from the suggestions `course` and `section` for GC1. For GC_3 the designer chose the more significant name `university_member` over the generic proposed name `person`.

3) ***Meaning choice.*** For each GC, the system proposes a meaning derived from the union of the meanings for the local class names GC_B. The designer could change this set by removing some meanings or by adding other ones. This is a crucial operation because assigns a universally-understandable meaning to each global class, i.e. to each item of the ontology.

4) ***Global attribute annotation***. We use the same approach for assigning names and meanings to attributes of a global class GC. For a given global attribute GA of a global class GC, we consider the set of local attributes, that MOMIS maps into the global attribute GA on the basis of the mapping table MT; this set is denoted by LA_{GA} (Local Attributes mapped to GA) and is defined as:

$$LA_{GA} = \{ LA \mid \exists LC \in GC, LA \in LC \land MT[GA][LA] \neq null \}$$

The set of Broadest Local Attributes mapped to GA is denoted by BLA_{GA} and is defined as:

$$BLA_{GA} = \{ LA \in LGA \mid \neg\exists y \in LGA, (LA\ NT\ y) \lor (y\ BT\ LA)\}$$

On the basis of BLA_{GA}, the designer annotates the GA in the same manner as global classes. Moreover, according to mapping function previously described we might develop a specific policy to automatically select meanings. For example, if GA is obtained as the conjunction of LA_1 and LA_2, the automatically selected meanings could be a hypernymy meaning of both LA_1 and LA_2.

Table III. University GVV annotation.

Global Class (GC)	GC_1	GC_2	GC_3
Local Classes of GC	CS.Course, UNI.Section	CS.Publication, UNI.Article	CS.Professor, CS.Person, UNI.School_Member, UNI.Research_Staff, CS.Student
Broadest Global Class of GC (BLC_{GC})	CS.Course, UNI.Section	CS.Publication	CS.Person
Names	course or section	publication	University_Member
Meanings	course#1	publication#1	person#1

V. CONCLUSION

MOMIS supports the semiautomatic building and annotation of domain ontologies by integrating the schemas of information sources, such as Web documents. The MOMIS's framework is currently adopted in the Semantic Web Agents in Integrated Economies (SEWASIE) European research project (www.sewasie.org). SEWASIE aims at implementing an advanced search engine that enables intelligent access to heterogeneous data sources on the Web via semantic enrichment, providing the basis for structured secure Web-based communication. To achieve this goal, SEWASIE creates a virtual network based on Sewasie information nodes (SINodes), which consist of managed information sources, wrappers, and a metadata repository. SINodes metadata represent GVVs of the overall information sources that each manage. To maintain the GVV of a SINode, we are investigating two distinct aspects: the system overload in maintaining the built ontologies and the effects of inserting new sources that could modify existing ontologies. We are, in fact, developing a methodology to insert a new source that provides a way to extend previously created conceptualisations, rather than starting from scratch[1]. Moreover, in order to "adapt" the existing ontology to the new "context", our work is also focused on managing dynamics by including updating and deleting operations. Future work will be addresses to improving the annotation phase (Section III) by allowing the integration designer to face multilingual environments, that is adopting a MultiWordNet-like lexicon ontology [17]. MultiWordNet is a multiligual lexical database that extends WordNet 1.6. At beginning, MultiWordNet contained information about English and Italian words only, but it has been recently updated to include Spanish WordNet (http://www.lsi.upc.es/~nlp/). This means that its internal organization is flexible enough to include additional languages and provide an effective multilingual lexicon ontology.

ACKNOWLEDGMENTS

This work is supported in part by the EC's 5th Framework IST program through the SEWASIE project in the Semantic Web Action Line. The SEWASIE consortium includes the University of Modena and Reggio Emilia (Sonia Bergamaschi, coordinator), the Universities of Aachen RWTH (M. Jarke), Roma La Sapienza (M. Lenzerini, T. Catarci), Bolzano (E. Franconi) and the companies IBM (G. Vetere), Thinking Networks AG (C. Engels) and CNA (A. Tavernari) as user organizations.

REFERENCES

[1] D. Beneventano et al., "Synthesizing an Integrated Ontology", *IEEE Internet Computing, Special Issue on The Zen of the Web,* September/October 2003, pp. 42-51

[2] S. Bergamaschi et al., "Semantic Integration of Heterogeneous Information Sources", *Data and Knowledge Eng.,* vol. 36, no. 1, 2001, pp. 215-249

[3] M. Lenzerini, "Data Integration: A theoretical Perspective", in *Proc. 21st Symp. Principles of Database Systems (PODS),* ACM Press, 2002, pp. 233-246

[4] N. Guarino, "Formal Ontology in Information Systems", in *Int'l Conf. Formal Ontology in Information Systems (FOIS 98),* IOS Press, 1998, pp. 3-15

[5] S. Abiteboul et al., "Data on the Web: From Relations to Semistructured Data and XML", *Morgan Kaufmann,* 2000

[6] R. Baumgartner et al. "Visual Web Information Extraction with Lixto", in *Proc. Very Large Database Conf. (VLDB 01),* Morgan Kaufmann, 2001, pp. 119-128

[7] V. Crescenzi et al. "RoadRunner: Automatic Data Extraction From Data-Intensive Web Sites", *Proc. Special Interest Group on Management of Data Conf. (SIGMOD 02),* ACM Press, p. 624

[8] J. Myllymaki, "Effective Web Data Extraction with Standard XML Technologies", *Proc. 10th Int'l World Wide Web Conf,* ACM Press, 2001, pp. 689-696

[9] D. Beneventano et al. "ODB_QOptimizer: A Tool for Semantic Query Optimization in OODB", *Proc. Int'l Conf. Data Eng. (ICDE 97),* IEEE CS Press, 1997, p. 578

[10]S. Castano et al. "Global Viewing of Heterogeneous Data Sources", *IEEE Trans. Data and Knowledge Eng.,* vol. 13, no. 2, 2001, pp. 277-297

[11]B. Everitt, Cluster Analysis, Heinemann, 1974

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

[13]M. Klein et al. "Ontology Versioning on the Semantic Web", *Proc. 1st Int'l Semantic Web Working Symp.,* Stanford Univ. Press, 2001, pp. 75-91

[14]R. Baeza-Yates and B. Ribeiro-Neto, "Modern Information Retrieval", Addison Wesley Longman, 1999.

[15]D. Beneventano et al. "Consistency Checking in Complex Object Database Schemata with Integrity Constraints", *IEEE Trans. Knowledge and Data Eng,.* Vol. 10, no. 4, 1998, pp. 576-598.

[16]D. Beneventano et al. "Description Logics for Semantic Query Optimization in Object-Oriented Database Systems", *ACM Trans. Database Systems,* vol. 28, no. 1, 2003, pp. 1-50.

[17]E. Pianta et al. "Developing an aligned multilingual database", *Proc. 1st Int'l Conference on Global WordNet,* 2002, http://multiwordnet.itc.it/.