Building an Ontology with MOMIS*#

Domenico Beneventano[1,2], Sonia Bergamaschi[1,2], Francesco Guerra[1]

DII - Università di Modena e Reggio Emilia
Via Vignolese 905 - Modena
{lastname.firstname}@unimo.it

[2]IEIIT-CNR Istituto di Elettronica e di Ingegneria dell'Informazione
e delle Telecomunicazioni
Viale Risorgimento 2 – Bologna

1. Introduction

Nowadays the Web is a huge collection of data and its expansion rate is very high. Web users need new ways to exploit all this available information and possibilities. A *new vision of the Web*, the Semantic Web[1], where resources are annotated with machine-processable metadata providing them with background knowledge and meaning, arises. A fundamental component of the Semantic Web is the ontology; this "*explicit specification of a conceptualization*" [6] allows information providers to give a understandable meaning to their documents.

MOMIS (Mediator envirOnment for Multiple Information Sources) [3] is a framework for information extraction and integration of heterogeneous information sources. The system implements a semi-automatic methodology for data integration that follows the *Global as View* (GAV) approach [11]. The result of the integration process is a global schema, which provides a reconciled, integrated and virtual view of the underlying sources, GVV (Global Virtual View). The GVV is composed of a set of (global) classes that represent the information contained in the sources. In this paper, we focus on the MOMIS application into a particular kind of source (i.e. web documents), and show how the result of the integration process can be exploited to create a conceptualization of the underlying domain, i.e. domain ontology for the integrated sources. GVV is then semi-automatically annotated according to a lexical ontology. With reference to the Semantic Web area, where generally the annotation process consists of providing a web page with semantic markups according to an ontology, we firstly markup the local metadata descriptions and then the MOMIS system generates an annotated conceptualization of the sources. Moreover, our approach "builds" the domain ontology as the synthesis of the integration process, while the usual approach in the Semantic Web is based on "a priori" existence of ontology.

2. The MOMIS system

In this section, we describe the information integration process for building the GVV of a web pages' set. The process is shown in Figure 1.

2.1 ODL*I3*

For a semantically rich representation of schemas and object patterns, MOMIS uses an object-oriented language called ODL*I3*, which is an evolution of the OODBMS standard language ODL. ODL*I3* extends ODL with the following relationships expressing intra- and inter-schema knowledge for the source schemas:

- SYN (synonym of) is a relationship defined between two terms t_i and t_j that are synonyms in every involved source.
- BT (broader terms) is a relationship defined between two terms t_i and t_j, where ti has a broader, more general meaning than t_j. The opposite of BT is NT (narrower terms).
- RT (related terms) is a relationship defined between two terms t_i and t_j that are generally used together in the same context in the considered sources.

By means of ODL*I3*, only one language is exploited to describe both the sources (the input of the synthesis process) and the GVV (the result of the process). The translation of ODL*I3* descriptions into one of the Semantic Web standards such as RDF, DAML+OIL, OWL is a straightforward process. In fact, from a general perspective an ODL*I3* concept

[1] http://www.w3.org/2001/sw

[#] A complete version of this work appears on IEEE Internet Computing's special "Zen of the Web" issue, Sep-Oct 2003, 42-51 corresponds to a *Class* of a the Semantic Web standard, and ODL$_{I3}$ relationships are translated into *properties* (in particular the ISA ODL$_{I3}$ relationships are *subclassof* in the Semantic Web standards).

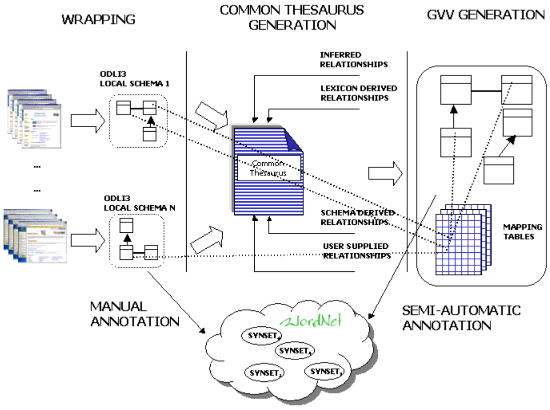

Figure 1: An overview of the ontology integration process

2.2 Wrapping: extracting data structure for sources

A wrapper logically converts the source data structure into the ODL$_{I3}$ information model. The wrapper architecture and interfaces are crucial, because wrappers are the focal point for managing the diversity of data sources.

For conventional structured information sources (e.g. relational databases), schema description is always available and can be directly translated. For semistructured information sources, a schema description is in general not directly available at the sources. A basic characteristic of semistructured data is that they are "self-describing" hence information associated with the schema is specified within data. Thus, a wrapper has to implement a methodology to extract and explicitly represent the conceptual schema of a semi-structured source. We developed a wrapper for XML/DTDs files. By using that wrapper, DTD elements are translated into semi-structured objects, according to different proposed methods [1], and in particular the OEM model [12].

Information is available on the Web mainly in HTML pages that are human-readable but cannot easily be automatically accessed and manipulated. In particular, HTML language does not separate data structure from layout. Thus, we need a further preliminary step of extraction: by means of Lixto [5], we translate the content of a web page (data and data structure) into a XML file, then we exploit the previously developed wrapper XML/DTD to acquire the source descriptions.

2.3 Running example

We consider how to build an ontology from two web sources related to the University domain. By means of a Lixto generated wrapper, the source content is translated into XML files according to the DTDs sketched in Table 1.

University Site (UNI)

```
 <!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)> ...
```

Computer Science Site (CS)

```
 <!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,
journal, editor)>
<!ELEMENT School_Member(name, e-mail)>
<!ELEMENT name (#pcdata)>...
```

Table 1: A fragment of the University (UNI) and Computer Science (CS) DTDs

By means of the XML/DTD wrapper, the obtained DTDs are translated into ODL$_{I3}$ descriptions. An example of the classes obtained in this step is shown in Table 2.

2.4 Annotation of a local source with WordNet

The WordNet database [10] contains 146,350 lemma organized in 111,223 synonym sets. WordNet's starting point for lexical semantics comes from the conventional association between the forms of the words - that is, the way in which words are pronounced or written - and the concept or meaning they express. These associations give rise to several properties, including synonymy, polysemy, and so forth. The correspondence between the words form (F_i) and their meaning (M_j) is synthesized in the so-called Lexical Matrix LM, where the element Lmij is true if the word form F_i can be used to express word meaning M_j.If $LM_{i,1}$, …, $LM_{i,k}$ k>1 are true, then the word form F_i is polysemous (i.e. it can be used to represent more than one meaning, M_1, …, M_k); if $LM_{1,j}$, …, $LM_{p,j}$ p>1 are true, then the word form $F_{i, \ldots,}$ Fp are synonyms.

```
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 < Publication > publication;}

Interface Publication
(Source Sc_site.dtd)
{ attribute string title;
  attribute string journal;
  attribute string editor }
```

Table 2: A piece of the University (UNI) and Computer Science (CS) sources in ODL$_{I3}$

The integration designer has to manually choose the appropriate WordNet meaning for each element of the conceptual schema. The annotation phase is composed of two different steps:

- **Word Form choice.** In this step, the WordNet morphologic processor aids the designer by suggesting a word form corresponding to the given term.
- **Meaning choice.** The designer can choose to map an element on zero, one or more senses.

This phase assigns a name, LEN (this name can be the original one or a word form chosen from the designer), and a set (eventually empty) of meanings, LEM_i (a class or attribute meaning is given by the disjunction of its set of meanings), to each local element (class or attribute) LE of the local schema:

$$LE = <LEN, \{LEM_1, \ldots , LEM_k \}>, k \geq 0$$

For example:

```
CS.Course = < course, {course#1} >
```

where Course#1 = 'education imparted in a series of lessons or class meetings'

In order to improve the accuracy of local source annotations with WordNet, we are evaluating how to extend WordNet. If a source description element (i.e. a class or an attribute name) has no correspondent in the reference lexical ontology (WordNet in our case), the designer may add a new meaning and proper relationships to the existing meanings

2.5 Common Thesaurus Generation

MOMIS constructs a Common Thesaurus describing intra and inter-schema knowledge in the form of SYN, BT, NT, and RT relationships. The Common Thesaurus is constructed through an incremental process in which relationships are added in the following order:

1. *schema-derived relationships*: relationships holding at intra-schema level are automatically extracted by analyzing each schema separately. For example, analyzing XML data files, BT/NT relationships are generated from couples IDs/IDREFs and RT relationships from nested elements.
2. *lexicon-derived relationship:* we exploit the annotation phase in order to translate relationships holding at the lexical level into relationships to be added to the Common Thesaurus. For example, the hypernymy lexical relation is translated into a BT relationship.
3. *designer-supplied relationships*: new relationships can be supplied directly by the designer, to capture specific domain knowledge. If a nonsense or wrong relationship is inserted, the subsequent integration process can produce a wrong global schema;
4. *inferred relationships*: Description Logics techniques of ODB-Tools [2] are exploited to infer new relationships, by means of subsumption computation applied to a "virtual schema" obtained by interpreting BT/NT as subclass relationships and RT as domain attributes.

In our running example, some of the relationships automatically obtained by MOMIS and proposed at the integration designer are the following (the number denotes the kind of derivation of relationships):

```
1 CS.Professor NT CS.Person
2 UNI.Article NT CS.Publication
3 UNI.Research_Staff SYN
CS.Professor
4 UNI.Research_Staff NT
CS.Person
```

2.6 GVV generation

The MOMIS methodology allows us to identify similar ODL$_{I3}$ classes, that is, classes that describe the same or semantically related concept in different sources. To this end, *affinity coefficients* are evaluated for all possible pairs of ODL$_{I3}$ classes, based on the relationships in the Common Thesaurus properly strengthened. Affinity coefficients determine the degree of matching of two classes based on their names (*Name Affinity* coefficient) and their attributes (*Structural Affinity* coefficient) and are fused into the *Global Affinity* coefficient, calculated by means of the linear combination of the two coefficients [4]. Global affinity coefficients are then used by a hierarchical clustering algorithm, to classify ODL$_{I3}$ classes according to their degree of affinity.

For each cluster Cl, a **Global Class** GC, with a set of **Global Attributes** GA_1, …, GA_N , and a **Mapping Table** MT, expressing mappings between local and global attributes, are defined.

The Mapping Table is a table whose columns represent the local classes (LC), which belong to the Global Class and whose rows represent the global attributes. An element `MT[GA][LC]` is a function which represents how local attributes of LC are mapped into the global attribute GA :

`MT[GA][LC]= f(LAS)`

where LAS is a subset of the local attributes of LC.

Some simple and frequent cases of such function are the following:

- *identity*: LAS is a singleton, LAS = {LA}, and f is the identity function; in this way we express that the GA value is equal to the `LA` value; we denote this case as `MT[GA][LC] = LA`
- *constant*: `GA` assumes into LC a constant value set by the designer; we denote this case by `MT[GA][LC] = const`
- *undefined*: `GA` is undefined into `LC`; we denote this case as `MT[GA][LC] = null.`

The Global Class and Mapping Table generation is a synthesis activity performed interactively with the designer. A preliminary set of Global Attributes GA_1, …, GA_N and mappings are automatically generated, and proposed to the designer, as follows.

First, local attributes of the local classes belonging to GC are grouped on the basis of SYN and BT/NT relationships among local attributes.

Formally, let ↔ be a relation defined between two local attributes LA_1 and LA_2 as follows:

$LA_1 \leftrightarrow LA_2$ iff LA_1 SYN LA_2 or LA_1 BT LA_2 or LA_1 NT LA_2 is in the Common Thesaurus.

Let ⇔ be the equivalence relation defined as the transitive-reflexive-symmetric closure of ↔.

Given a local attributes LA, [LA] denotes the equivalence class of LA w.r.t. ⇔. Given a Global Class GC, we consider a Global Attribute GA, for each element of the set

$\{[LA] \mid LA$ is an attribute of LC and $LC \in GC\}$

For each element of the mapping table `MT[GA][LC]=f(LAS)` the proposed set LAS is the set of the attributes of the local class LC which belong to the equivalence class related to GA. This set can be:

- *empty*: GA does not have any representation in the local class LC: in this case the designer has to choose between the undefined (default) or the constant function;
- *a singleton* : the function f may represent the identity function (default), i.e. GA and LA represent the same information, or f is a translation function.

In our running example the clustering 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)
```

and, for Global2, the following Mapping Table, where all the maps are automatically produced except for $Const_1$ set by the designer, is generated.

	UNI.Article	CS.Publication
Title	Title	Title
Year	Year	Const_1 [2]
Journal	Journal	Journal
Conference	Conference	NULL
Editor	NULL	Editor

Table 4: Mapping Table of the global class Global2 (Publication)

3 Global Virtual View Annotation

In this section, we propose a semi-automatic methodology annotate a GVV, i.e. to assign a name, GEN, and a set (eventually empty) of meanings, GEM_i (a class or attribute meaning is given by the disjunction of its set of meanings) to each global element (class or attribute) GE:
$GE = <GEN, \{GEM_1, \ldots , GEM_p\}>, p \geq 0$

3.1 Global Class Annotation

In order to semi-automatically associate an annotation to each global class, we consider the set of all its "broadest" local classes, w.r.t. the relationships included in the Common Thesaurus, denoted by GC_B:

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

In our example:

	GC	GC_B
GC_1	CS.Course, UNI.Section	CS.Course, UNI.Section
GC_2	CS.Publication, UNI.Article	CS.Publication
GC_3	CS.Professor, CS.Person,UNI.School_Member, UNI.Research_Staff, CS.Student	CS.Person

On the basis of GC_B, the designer will annotate the global class GC as follows:

- **name choice**: the integration designer is responsible for the choice of the GC name: the system only suggests a list of possible names. The designer may select a name, i.e. a label to identify the GC, within the proposed list or select another name not belonging to the list.
- **meaning choice**: the union of the meanings of the local class names in GC_B are proposed to the designer as meanings of the Global Class. The designer may change this set, by removing some meanings or by adding other ones.

With respect to our example, the proposed annotations are the following:

GC	Names	Meanings
GC_1	course or section	course#1
GC_2	Publication	Publication#1
GC_3	University_Member	person#1

Table 5: University GVV annotation

3.2 Global Attributes Annotation

We extend the previously used approach for names and meanings of the attributes. Given a global attribute GA of the global class GC, we consider the set LGA of local attributes, which are mapped into GA:

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

and the set of all its ``broadest" local attributes, denoted by LGA_B:

$LGA_B = \{LA \in LGA \mid \neg\exists y \in LGA, (LA\ NT\ y)\}$

On the basis of LGA_B, the designer will annotate the global attribute as described for global classes. Moreover, according to mapping function, we may develop some specific policy to automatically select meanings.

4 Concluding remarks and future work

In this paper, we presented a methodology for supporting the semi-automatic building, annotation of a domain ontology obtained by integrating web documents with the MOMIS system. Some methodologies that aid the generation process of semantic mappings between data sources and mediated schema, starting from annotated schemas, have been presented; as pointed in [7], generating semantic

[2] For example, in order to specify all the publications of CS source are published on 2003, the designer may set Const_1=2003

mappings is a current challenge in data integration. In this paper, we do not take into account problems arising when two o more schemas are merged [13].

The annotated ontology may be exploited to support dynamics issues, i.e. to have ontology consistent with the domain that refers to. Many interesting solutions have been developed with regard to this topic [8,9] and an outstanding idea is to exploit multiple variants of the same ontology to cope with changes. This approach, called o*ntology versioning*, is different from our idea where a single ontology has to be kept consistent with the sources, which refer to. So, if new sources are added/deleted, or if some changes occur in the sources, the corresponding GVV has to change. In order to restart the integration process from scratch, we are developing a methodology for integrating a new source, which exploits the previous integration work, i.e., a built-up GVV, without restarting the integration process from scratch.

The Momis methodology is currently adopted in the Sewasie (Semantic Web Agents in Integrated Economies) European research project (www.sewasie.org). Sewasie's goal is to design and implement an advanced search engine that enables intelligent access to heterogeneous data sources on the Web via semantic enrichment that provides the basis for structured secure Web-based communication. To achieve this goal, Sewasie realizes a virtual network, whose nodes are Sewasie Information Nodes (SINode). SINodes are mediator-based systems that represent a virtual view of the overall information managed within any SINode and consists of the managed information sources, wrappers, and a metadata repository. We think that the methodology implemented in Momis could be exploited to create the kernel of an SINode

Bibliografy

[1] S. Abiteboul, P. Buneman, and D. Suciu. "Data on the Web - From Relations to Semistructured Data and XML". Morgan Kaufmann, 2000.

[2] D. Beneventano, S. Bergamaschi, C. Sartori, M. Vincini "ODB-QOptimizer: a tool for semantic query optimization in OODB." Int. Conference on Data Engineering ICDE97, UK, April 1997.

[3] S. Bergamaschi, S. Castano, D. Beneventano, M. Vincini: "Semantic Integration of Heterogeneous Information Sources", DKE, Vol. 36, Num. 1, Pages 215-249, Elsevier Science B.V. 2001.

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

[5] R. Baumgartner, S. Flesca, G. Gottlob: Visual Web Information Extraction with Lixto. VLDB 2001: 119-128

[6] T. R. Gruber. A translation approach to portable ontologies. Knowledge Acquisition, 5(2):199-220, 1993

[7] A. Halevy Data Integration: a Status Report. Proceedings of the German Database Conference, BTW-03

[8] J. Heflin, J. Hendler Dynamic Ontologies on the Web. In Proceedings of the Seventeenth National Conference on Artificial Intelligence (AAAI-2000). AAAI/MIT Press, CA, 2000. pp. 443-449.

[9] M. Klein, D. Fensel Ontology Versioning on the Semantic Web. In 1th Semantic Web Working Symposium. 2001.

[10] A.G. Miller. A lexical database for English. Communications of the ACM, 38(11):39:41,1995

[11] M. Lenzerini Data Integration: A Theoretical Perspective. PODS 2002: 233-246

[12] Y. Papakonstantinou, H. Garcia-Molina, J. Widom. Object exchange across heterogeneous information sources. In Proc of ICDE95, Taiwan, 1995

[13] R.A. Pottinger, P. A. Bernstein: Merging Models Based on Given Correspondences, University of Washington Technical Report UW-CSE-03-02-03. February 2003.