BUILDING A VIRTUAL VIEW OF HETEROGENEOUS DATA SOURCE VIEWS

Lerina Aversano, Roberto Intonti, Clelio Quattrocchi and Maria Tortorella
Department of Engineering, University of Sannio, via Traiano 82100, Benevento, Italy

Keywords: Schema Matching, Schema Merging, Virtual View, Heterogeneous Data Source.

Abstract: In order to make possible the analysis of data stored in heterogeneous data sources, it could be necessary a preliminary building of an aggregated view of these sources, also referred as virtual view. The problem is that the data sources can use different technologies and represent the same information in different ways. The use of a virtual view allows the unified access to heterogeneous data sources without knowing details regarding each single source. This paper proposes an approach for creating a virtual view of the views of the heterogeneous data sources. The approach provides features for the automatic schema matching and schema merging. It exploits both syntax-based and semantic-based techniques for performing the matching; it also considers both semantic and contextual features of the concepts. The usefulness of the approach is validated through a case study.

1 INTRODUCTION

The diffusion of information systems and continuos development of communication technologies allows accessing numerous data sources, and offers the possibility of extracting and analyzing the available information. However, if a user needs to analyse data of the same domain but stored in different data souces, he could find difficulties for synthesizing the information useful to his purpose. These difficulties increase if the source are heterogeneous, because they are created in different times, on different systems and with different criteria. Forms of heterogenity may exist in the *technology* used for the data source implementation, and in the adopted *modeling* formalism. The first kind of heterogeneity is due to the type of technology used for building the data source, as different DBMSs and persistence models can be used. The modeling heterogeneity is due to the schema describing the real data, as two different sources can use different schemas for representing the same set of information. This is mainly due to the different choises of the data source designers, as a standard method for modeling concepts doesn't exist.

For all the reasons above, the management and analysis of data stored in heterogenous data sources represents a critical problem. Therefore, the support of a process for building an aggregated view of a set of heterogenuos data sources could be useful. The aggregated view is also referred as virtual view and allows the unified access to heterogeneous data sources as if they would be a unique source. The cited process should assure the maximun trasparency to the end users and maximun autonomy for managing the involved data sources. If the number of sources is low and data model is not too complex, it is possible to adopt a manual process aiming at creating an homogeneous model considering all the data, otherwise the manual approach is not feasible.

This paper proposes an aggregation process aiming at creating a virtual view of the schemas of a set of data sources to be analysed. The virtual view provides a unique vision of the real data stored in different data sources. The process foresees a progressive contruction of the virtual view. In fact, if a new data source is acquired the virtual view is updated so that the new data source is accessible through it. The acquisition of a new data source is performed in two phases: *Schema Matching*, generating the mapping among the elements of the two schemas; and *Schema Merging*, during which the merge of the schemas is performed. The solutions existing in literature are focused only on specific aspects of the problem or require a level of interaction with an expert user. The use of semantic information allows some of these solutions reaching

Aversano L., Intonti R., Quattrocchi C. and Tortorella M. (2010).
BUILDING A VIRTUAL VIEW OF HETEROGENEOUS DATA SOURCE VIEWS.
In *Proceedings of the 5th International Conference on Software and Data Technologies*, pages 266-275
DOI: 10.5220/0003011702660275

good results in a strongly heterogenous context. But a complete solution detecting and solving all the heterogeneous forms does not currently exist. The approach proposed in this paper aims to detect and solve the heterogeneous forms that are introduced from different data sources designer.
The rest of the paper is organized as follows: Section 2 discusses some related work; Section 3 describes the proposed approach; Section 4 introduces a case study for showing the results gotten by using the approach. Section 5 contains conclusive considerations.

2 RELATED WORK

Many researchers are involved in studying methodological and technological approaches for the aggregation of heterogeneous data sources. The main problem to be faced by these approaches is the identification and resolution of conflicts that exist between the schemas of the different data sources. In (Lee, 2003), three types of conflicts are defined: *nominal*, *structural* and *type*. The *nominal* conflicts are referred to both synonyms, i.e different terms used for indicating the same concept, and homonyms, a unique term employed for representing different concepts. A *structural* conflict is introduced when different structures are used for representing the same concept. Finally, a *type* conflict exists when the same concept is modelled by using different data types. The activities mainly considered in literature for merging different schemas are the following two: *Schema Matching* and *Schema Merging*. The two following subsections discuss the approaches of Schema Matching and Schema Merging presented in the literature, while the subsequent subsection compares the different discussed approaches.

2.1 Schema Matching

The schema matching is a process for searching similar or equivalent elements existing between two schemas (Denivaldo, 2006) (Rahm, 2001) (Jayant, 2001). The result of this process is a set of mappings identifying the corresponding elements of the two schemas. The mapping can be generated by using *syntax-based* (Cohen, 2003) and/or *semantic-based* techniques (Giunchiglia, 2003). The former kind of technique analyses the syntactic characteristics for determining if two elements are equivalent. They return a factor belonging to the range [0,1]. Two elements are considered equivalent if the returned factor is greater than a given threshold. The semantic-based techniques extract the semantic relationships existing among the concepts of the real world that the two considered elements represent.

Cupid (Jayant, 2001) is a software component executing a schema matching activity. It uses a thesaurus for identifying acronyms, homonyms and synonyms of the terms in the schemas to be compared. It generates a mapping between the schemas of the two data sources by executing two phases: *Linguistic Matching* and *Structure Matching*. The Linguistic Matching generates the mapping regarding names, data types and domains of the elements. The Structure Matching regards the context of the elements and is based on the idea that two elements are structurally similar if their composing elements are equivalent.

Given two schemas represented as trees, **S-Match** (Giunchiglia, 2003), (Giunchiglia, 2004a), (Giunchiglia, 2004b), generates a set of semantic relationships performing element-level and structure-level matching. Semantic relationships among two single elements are generated by using WordNet (Miller, 1995), that is a lexical database for the English language. Given a word and the relative syntactic category (i.e., noun, verb, adjective and adverb), the system returns a set of *sense,* or *synset*. A sense is the meaning/concept that a word has/represents in the real world. Every sense has a *gloss*, that is a textual description of the meaning. WordNet is a real world ontology since senses are organized in a semantic net. The main stored semantic relationships are: synonymy, type of, is part of. A semantic matching among two elements is generated by extracting the senses and verifying if a relationship exists for at least one couple of senses. If it exists, the relation is returned.

GLUE (AnHai, 2003) is the only analyzed system that performs the matching by using machine learning techniques. It accesses instances for determining the type of relationship existing between two concepts.

The schema matching performed by **Puzzle** (Huang, 2005) automatically generates the 1:1 mapping by two phases: *Linguistic Matching* and *Contextual Matching*. The Linguistic Matching generates a similarity factor for every couple of classes by only considering the class names. For this purpose, it combines both syntax-based and semantic-based techniques. The Contextual Matching generates a similarity factor by considering properties and relations of classes. By combining the obtained values Puzzle generates for every couple of classes one of following relation:

Table 1: Comparison among Approaches of Schema Matching and Schema Merging.

	Cupid	Dike	S-Match	Puzzle	PROMPT	GLUE	MOMIS
SCHEMA MATCHING							
SUPPORTED SCHEMAS	XML Relational	ER	Ontology	Ontology	Ontology	Ontology	Any Schema
LEVEL							
SCHEMA	✓	✓	✓	✓	✓		✓
INSTANCE						✓	
GRANULARITY							
ELEMENT LEVEL	✓	✓	✓	✓	✓	✓	✓
STRUCTURE LEVEL	✓	✓	✓	✓	✓	✓	✓
MATCHER ELEMENT-LEVEL							
SYNTAX-BASED	✓	✓	✓	✓	✓	✓	✓
SEMANTIC-BASED	✓		✓	✓			
TYPE OF GENERATED MAPPING S	$/\subseteq/\supseteq$	$/\subseteq/\supseteq/$	$/\subseteq/\supseteq$	$/\subseteq/\supseteq$	=	$/\subseteq/\supseteq$	=
CARDINALITY	1:1/1:n	1:1	1:1	1:1	1:1	1:1	1.1
AUTOMATION LEVEL							
SEMI-AUTOMATIC					✓		✓
AUTOMATIC	✓	✓	✓	✓		✓	
SCHEMA MERGING							
AUTOMATION LEVEL							
SEMI-AUTOMATIC					✓		
AUTOMATIC		✓		✓			✓
APPLICATION AREA	Schema Matching	Integration of heterogeneous database	Schema Matching	Integration of heterogeneous ontologies	Integration of heterogeneous ontologies	Integration of heterogeneous ontologies	Integration of heterogeneous database

subclass, *superclass*, *equivalentclass*, *sibling*, *other*.

PROMPT performs the schema matching by adopting a semi-automatic approach and applying syntax-based techniques (Fridman Noy, 2000). **DIKE** generates relations of synonymy, homonymy, is-a, overlap by applying syntax-based techniques and exclusively analyzing the contextual characteristics of the elements (Ursino, 2003). **Momis** is based on the affinity factor existing among the classes by considering their names and attributes. The system is able to detect only equivalence relations (Bergamaschi, 1997).

2.2 Schema Merging

The last four approaches discussed in the previous subsection, Puzzle, PROMPT, DIKE and Momis, perform also the Schema Merging activity. This activity (Lee, 2003) (Fong, 2006) (Chiticariu, 2008) (Hyunjang, 2005) performs the merging of two schemas, given the mappings produced by the schema matching, after their validation of a user.

Puzzle (Huang , 2005) performs automatic merging of heterogeneous ontologies by using the relations generated by the schema matching.

As Puzzle, **PROMPT** (Fridman Noy, 2000) performs the merging of heterogeneous ontologies, but, in this case, a semi-automatic approach is adopted. For every mapping, PROMPT generates a set of merge operations to be performed. Once the user selects the operation, the system performs it and display new suggestions and possible conflicts that the user must resolve. The types of conflicts that can be created are: name conflicts, leaning references, redundancy in a is-a hierarchy.

DIKE (Ursino, 2003) performs the merging of Entity Relationship schemas. Given the relations generated by the schema matching activity, schemas are grouped into clusters. The schemas belonging to the same cluster are integrated in a virtual schema. This process is iterated on the produced schemas and finishes when one schema is obtained.

On the base of the factors calculated in the schema matching activity, **Momis** (Bergamaschi, 1997) groups concepts in clusters. For each cluster, only one class will be defined in the new virtual schema.

2.3 Comparison

Table 1 compares the approaches discussed in the previous two subsections. Some observation will be reported in the following with reference to the table content.

Table 1 highlights that Cupid, Puzzle and S-Match adopt the best approaches with reference to schema matching activity. In fact, they integrate semantic-based and syntax-based techniques.

Regarding the types of mapping, Momis and PROMPT are able to only identify the equivalence relationship. Both of them adopt a semi-automatic approach, since the generated mapping depends on the choices that the user makes during the schema matching activity.

Puzzle considers the best approach for the integration of heterogeneous databases, since it generates the mapping by combining both contextual and semantic characteristics of concepts. Moreover, it performs the merging in an automatic way. A disadvantage of Puzzle is represented by the fact that it is only able to produce mappings 1:1.

PROMPT requires a degree of iteration with the user that can be just permitted for databases with small dimensions.

DIKE and Momis are the only tools supporting the integration of heterogeneous databases. Their problem regards the fact that both of them produce mapping exclusively by analyzing the structures of the schemas without using auxiliary information, such as thesauri, dictionaries, and so on.

The approach proposed in this paper tries to overcome the limitations introduced by the listed approaches. It considers both Schema Matching and Schema Merging activities and uses both syntax-based and semantic-based techniques. In addition, it foresees the automatic support of the full process of generation of a complete virtual view of the analysed set of heterogeneous data sources.

3 PROPOSED APPROACH

The proposed aggregation process aims at providing a virtual view of the data stored in heterogeneous sources. Figure 1 shows a high-level view of the proposed approach. The virtual view is created through an incremental merging process of the local schemas of the data sources to be acquired. The advantage of using a virtual view is offering a uniform access to the single data sources from the external software applications.

Figure 2 shows the relationships between the two main components of the proposed solution. The *mediator* assures the maximum transparency to the end users, since it coordinates the data flow among the local database and applications. In particular, the applications perform the queries with reference to the virtual view, and the *mediator* converts these queries into simpler ones referred to the single data sources. The *wrappers* are the software components that directly interact with the respective local databases. They perform the following operations: translation of the local schemas into a global language; sending of the queries to data sources; collection of the query results and sending them to the mediator. The *wrappers* allow the acquisition of any source independently from the used technology.

The full approach, creating and updating a virtual view, is shown in Figure 3. It receives as input the schema of the data source to be acquired and produces as output the new virtual view. The approach consists of the following five main activities:

Pre-processing: it performs a first analysis and processing of the input data source to be acquired. It is composed of the following three tasks:

- *Schema Extraction*: it extracts the local schema from the data source and represents it as an object model, composed of classes, properties, and relationship *is-a* and *has*.
- *Tokenization*: it decomposes the names of the classes and properties into tokens through the recognition of special characters.
- *POS Tagging*: it associates every token to its lexical category. The output of the task is an Element List, that is a list associating every class and property to the corresponding tokens, and each token to the related lexical category.

Sense Mapping: it associates the needed semantic information for applying the semantic-based matcher to the tokens of the Element List. The semantic information is collected from a lexical semantic database received as input. The used database is WordNet (Miller, 1995). It groups the tokens with similar meaning on the basis of their lexical category and memorizes their semantic relationships.

The activity of Sense Mapping is composed of the following two tasks:

- *Sense Extraction*: it accesses the semantic database and associates each token with the senses related to its lexical category, determined in the POS-tagging task of the Pre-Processing activity;
- *Sense Filtering*: it uses genetic algorithm based on the Similarity package of WordNet for selecting the correct sense for every token, and filters the other ones. The Similarity package of WordNet includes a set of measures using the structure of WordNet for determining the similarity degree of two senses (Pedersen, 2004) (Pattwardhan, 2003).

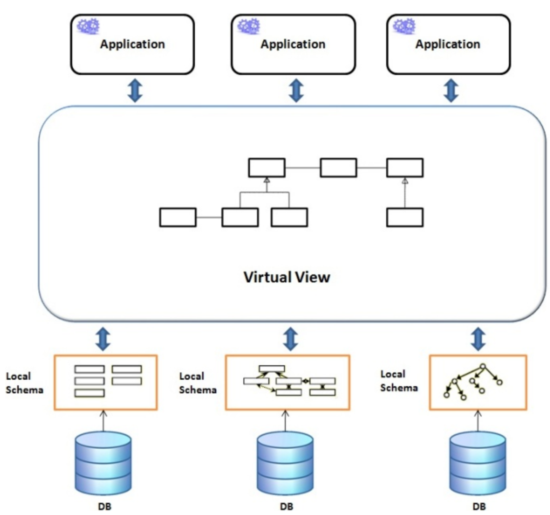

Figure 1: The Virtual View obtained by heterogeneous data sources.

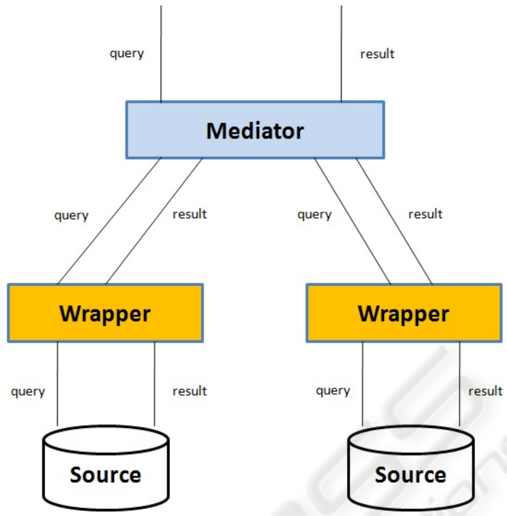

Figure 2: Details of the Mediator/Wrapper components.

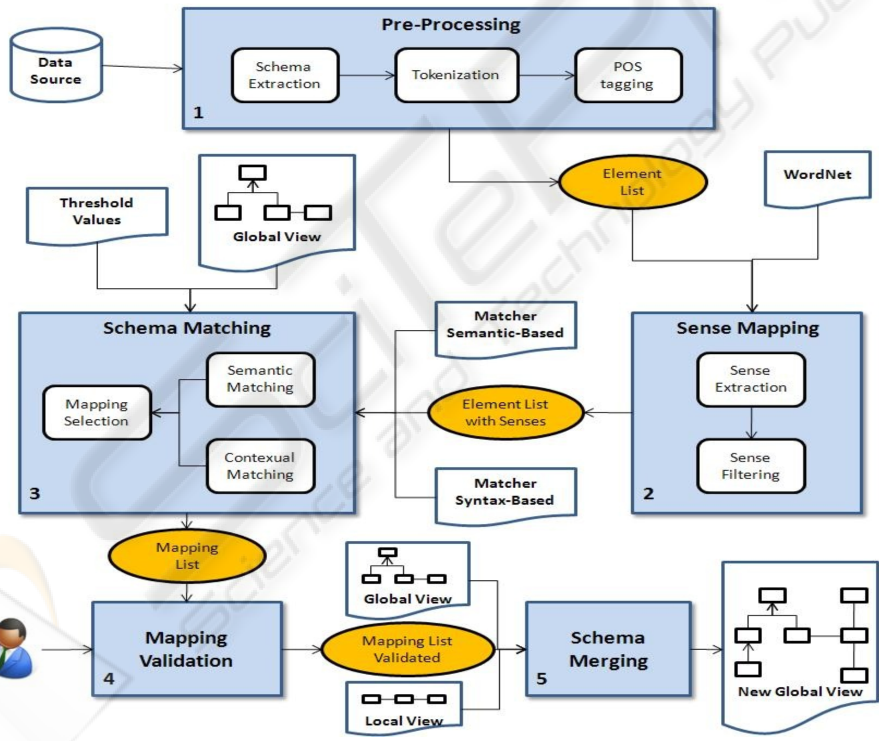

Figure 3: Overall view of the proposed approach.

Schema Matching: it generates a set of mapping among the classes of the virtual view and the local schema, combining both semantic and contextual characteristics. The Schema Matching activity is composed of the following three tasks:

– *Semantic Matching*: it exclusively considers the semantic characteristics of the classes. The matching is performed by considering the objects of the real world that the classes represent in the belonging schema. For every couple of classes (C_V, C_L), formed of one class of the virtual view and one of the schema of the local data source, a semantic-based matcher is used for determining the semantic relationship existing among the classes of the couple, indicated with *SemanticRel*. Given the senses associated to the tokens of the classes, the matcher accesses WordNet and checks if an equivalence (=) or is-a ($\subseteq / \supseteq$) relationship exists for at least one couple of senses. In the affirmative case, the found relationship is returned, otherwise idk (the don'ts known) is returned. A factor, called *semanticSim*, is also associated to the relationship for indicating the degree of existing semantic relation in the case the name of the classes is composed of more than one token.

– *Contexual Matching*: it considers the contextual characteristics of the classes and the way they are modelled for calculating the similarity degree existing between each couple of classes or aggregation of them. Actually, the greater the number of the equivalent properties among two classes is, the higher their similarity degree. First, the mapping among the properties is produced by applying a semantic-based or syntax-based matcher. Then, for each couple of classes (C_V, C_L), the *ContextualSim* is calculated. It is a coefficient belonging to the range [0,1], evaluated by applying the Jaccard's metric to the properties of the two classes (Tan, 2005). Let $P(C_V)$ and $P(C_L)$ be the sets of the properties of C_V and C_L, respectively, the Jaccard's metric is evaluated as the rapport between the number of the common properties of the two classes and the total number of properties:

$$ContextualSim(C_V, C_L) = \frac{\#(P(C_V) \cap P(C_L))}{\#(P(C_V) \cup P(C_L))}$$

– *Mapping Selection*: it generates the mappings 1:1, 1:n, n:1, n:m, by combining the results of previous activities and using some threshold values received in input. The idea is that, if a semantic relationship, SemanticRel, exists between a set of classes, and the degree of contextual similarity, ContexualSim, is greater or equal than a given threshold value, the corresponding mapping can be considered valid. If the relation is equal to the threshold value, it is indicated with the symbol $\alpha_{=}$, else $\alpha_{\subseteq/\supseteq}$ is used. Lowering the threshold values, a major relevance is given to the semantic characteristics than to the contextual ones. The task produces two lists: one regarding the so-called automatic mapping, that can surely be considered as valid and does not need validation; and one including semi-automatic mappings that need to be validated. A mapping is automatic if the two following conditions are satisfied: (i) it concerns two classes connected by a semantic relation with the higher *SemanticSim* value; (ii) a Jaccard factor equal to 1 is associated to the classes involved in the mapping, meaning that a 1:1 correspondence exists between the related classes properties.

Mapping Validation: it permits the user to validate and modify the automatically generated mapping.

Table 2: Schema Merging Algorithm.

Step 1.	Create **NewMappingList** and new schema **NewGlobalView**
Step 2.	For each *Mapping($\{C_G\}_k$, $\{C_L\}_z$)* execute a **Merge** operator.
Step 3.	Insert classes and properties of **LocalView** that are not present in **NewGlobalView**
Step 4.	Delete redundancy relations from **NewGlobalView**.
Step 5.	Execute *refactoring* of **NewGlobalView**
Step 6.	Generate the mapping file for the **LocalView**
Step 7.	**NewGlobalView** is the new virtual view

Schema Merging: it performs the merging of local schema in the virtual view for generating a new virtual view. The new virtual view must satisfy the requirement of not redundancy and completeness, that is it must include all the information of the acquired schemas. The algorithm used for the schema merging is shown in Table 2.

Step 1 in Table 2 initializes the new virtual view with that first local data source to be considered. Step 2 applies some Merge operators for performing the merge of the classes included in the current mappings. The execution of Steps 3 and 4 aims at guaranteeing completeness and not redundancy of the new virtual view. Step 5 executes the *refactoring*, that is a process evolving the new virtual view assuring correctness and minimality. Step 6 produces the new file of the mapping between the new virtual view and local data sources. This file adds the mappings between the new virtual view and the schema of the acquired data source and updates the mappings with the schemas of the data sources previously acquired.

4 CASE STUDY

This section describes the application of the proposed approach to a case study, and considers the data sources used in the health care domain.

The initial virtual view is built starting from the first data source, shown in Figure 4. The schema of the second data source to be acquired is shown in Figure 5. Its acquisition required the updating of the virtual view, so that its data are uniformly accessible through it.

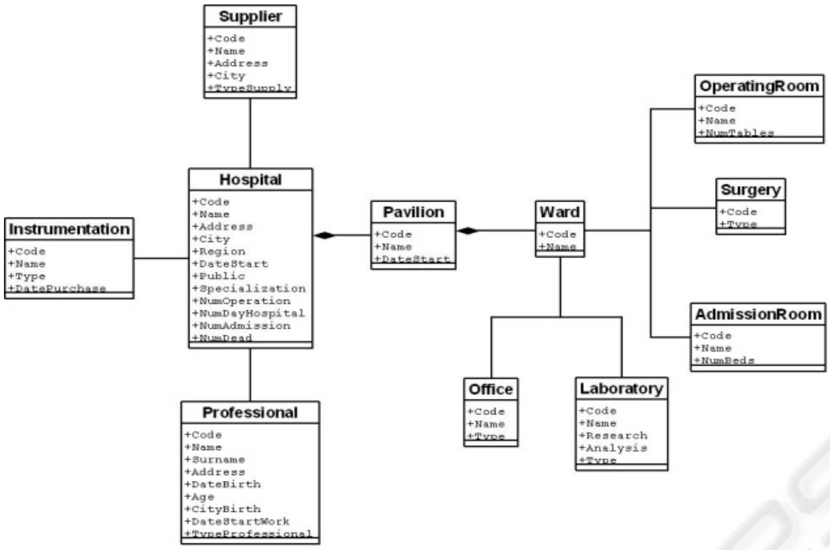

Figure 4: Initial virtual view obtained from the first data source.

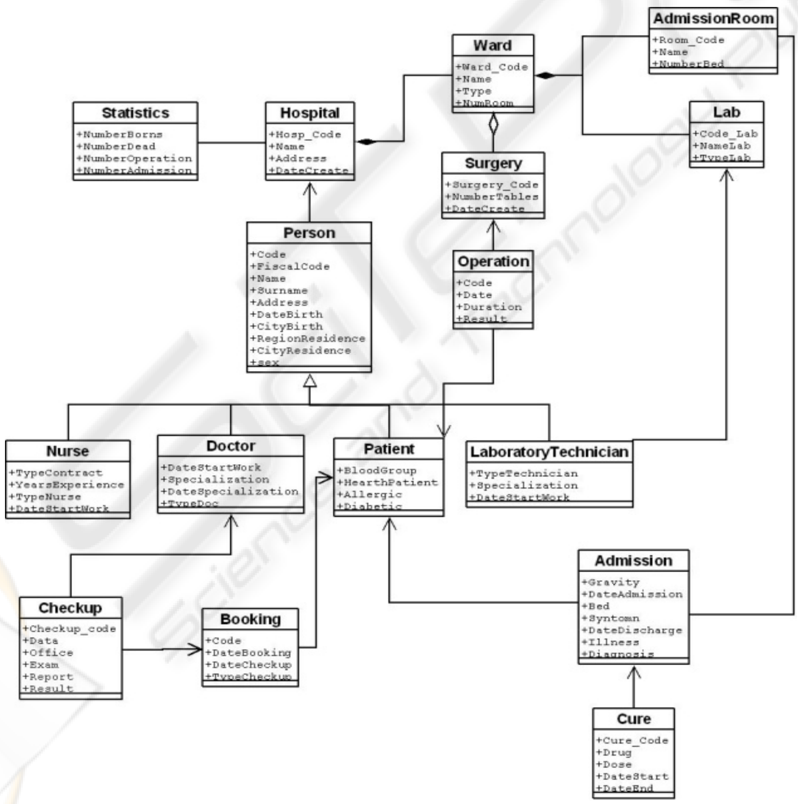

Figure 5: Local Schema of the second data source.

Analyzing the two schemas in Figures 4 and 5, both nominal and structural conflicts emerged. As an example, there is a *nominal conflict* with reference to the name `Surgery`. Indeed, it is used in the two considered schemas for representing different concepts. In the virtual view, it represents a room where a doctor can be consulted, while it represents an operating room in the second local view. Moreover, as an example of *structural conflict*, the attributes of the class `Statistics` in the local schema are modeled as attributes of the class `Hospital` in the global schema.

The application of the approach steps of the proposed approach for acquiring the schema of the second data source depicted in Figure 5 is detailed in the following.

Pre-processing: the Schema Extraction activity takes out the schema of the second data source. The tokenization activity follows for extracting the tokens from the class names. As an example, tokens `Admission` and `Room`, and `Laboratory` and `Technician` are respectively identified from the classes `AdmissionRoom` and `Laboratory Technician`. Then, the POS Tagging activity associates the lexical category *Noun* to each token.

Sense Mapping: this activity associates the senses encoded in WordNet to the tokens gotten in the previous phase. For instance, the senses of the `Hospital` token are the following:

1. *Sense#1***:** a health facility where patients receive treatment.
2. *Sense#2*: a medical institution where sick or injured people are given medical or surgical care.

Table 3 shows the senses selected by the *Sense Filtering* task for some tokens.

Table 3: Sense Filtering Output.

CLASS	TOKEN	SENSE
`Hospital`	`Hospital`	**Sense#1**: a health facility where patients receive treatment.
`Ward`	`Ward`	***Sense#3:*** *block forming a division of a hospital (or a suite of rooms) shared by patients who need a similar kind of care*

Schema Matching: for performing this activity, it is first necessary to fix the threshold values.
In particular, the values adopted in the proposed case study are the following:

$\alpha_{=}$: 0.4 $\alpha_{\subseteq/\supseteq}$: 0.2 β: 0.8

where $\alpha_{=}$ is the threshold adopted for the equivalence relationship, $\alpha_{\subseteq/\supseteq}$ is the threshold adopted for the specialization/generalization equivalence relationship, and β is the threshold considered for selecting the correspondences found by using the Jaccard coefficient.

The correct mapping identified by the Schema Matching activity are the following:

1. $\mathtt{Lab}_L = \mathtt{Laboratory}_V$
2. $\mathtt{Ward}_L = \mathtt{Ward}_V$
3. $\mathtt{AdmissionRoom}_L = \mathtt{AdmissionRoom}_V$
4. $\mathtt{Surgery}_L = \mathtt{OperatingRoom}_V$
5. $[\mathtt{Hospital}_L, \mathtt{Statistics}_L] = \mathtt{Hospital}_V$
6. $\mathtt{Person}_L \supseteq \mathtt{Professional}_V$
7. $\mathtt{Nurse}_L \subseteq \mathtt{Professional}_V$
8. $\mathtt{Doctor}_L \subseteq \mathtt{Professional}_V$
9. $\mathtt{Person}_L \supseteq \mathtt{Supplier}_V$
10. $\mathtt{LaboratoryTechnician}_L$ Idk $\mathtt{Professional}_V$

The mapping related to the $\mathtt{AdmissionRoom}_L$ and $\mathtt{AdmissionRoom}_V$ concepts can be automatically accepted. Indeed, the Jaccard coefficient value of the couple formed by the two classes is equal to 1.The Schema Matching activity is able to identify both the existing nominal and structural conflicts. For example, in the considered case study, thanks to the use of the semantic-based matcher, equivalence relationships are identified among the classes $\mathtt{Laboratory}_V$ and $\mathtt{Lab}_L$, $\mathtt{OperatingRoom}_V$ and $\mathtt{Surgery}_L$, respectively, although synonyms are used for them.

Moreover, the mapping is identified between $\mathtt{Hospital}_V$ and the classes $\mathtt{Hospital}_L$ and $\mathtt{Statistics}_L$.

Mapping Validation: the user must delete the mapping automatically generated that are not correct and modify the mapping among the classes "$\mathtt{Professional}_V$" and "$\mathtt{LaboratoryTechnician}_L$".

Schema Merging: this activity performs the merging of the local schema into the virtual view for generating a new virtual view. It executes the algorithm described in Table 2. The updated virtual view of the analyzed case study is shown in Figure 6. Table 4 shows a fragment of the XML file produced for mapping the acquired data source to the components of the new virtual view. The table shows that a `global-class` element is used for each class of the virtual view. The `global-class` element includes a son element for each attribute of the class to which it is referred. These attributes are indicated with the tags `global-attribute`. The mapping is, then, introduced for each mapped attribute in the local view and is indicated with the `mapping-rule` tag. As a example, Table 4 shows that the `Name` and `Surname` attributes of the virtual class `Person` are mapped to the `Name` and `Surname` attributes of the local class `Person`, and that the `CityResidence` attribute of virtual class `Doctor` is mapped to the `CityResidence` attribute of local class `Person`.

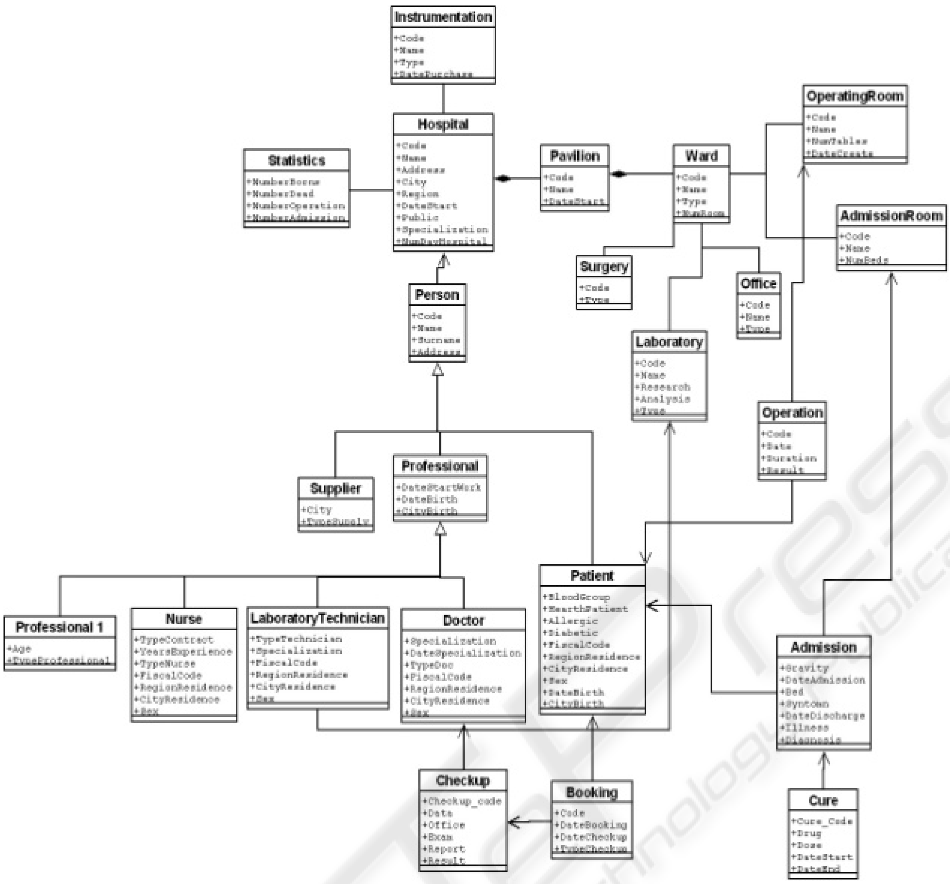

Figure 6: Updated Virtual View.

Table 4: File of mapping.

```
<global-class name="Person">
  <global-attribute name="Name">
    <mapping-rule>
      <attribute-ref class-refid="Person"
      attribute-refid="Name"/>
    </mapping-rule>
   </global-attribute>
   <global-attribute name="Surname">
   <mapping-rule>
     <attribute-ref class-refid="Person"
      attribute-refid="Surname"/>
    </mapping-rule>
   </global-attribute>
.............
</global-class>
<global-class name=‟Doctor">
  <global-attribute name="CityResidence">
    <mapping-rule>
      <attribute-ref class-refid="Person"
      attribute-refid="CityResidence"/>
    </mapping-rule>
   </global-attribute>
.............
</global-class>
```

5 CONCLUSIONS

This paper describes an approach proposed for creating and updating a virtual view of more than one heterogeneous data sources. The creation of a virtual view guarantees the access to more than one heterogeneous sources, as if they are a unique source. In the proposed approach, the virtual view is created through the merging of schemas containing the metadata of the single acquired data sources.

The solutions already proposed in literature concerning the aggregation of heterogeneous data sources, are focused just on specific aspects of the problem or require a too elevated level of interaction with the user. The proposed approach completely automates the activities of schema matching and schema merging. It just requires the intervention of the user for defining the threshold values and validating the identified mappings. In particular, the Schema Matching activity produces mappings of cardinality 1:1, 1:n, n:1, n:m, among the classes of two schemas by considering both the semantic and contextual aspects. Mapping among two single elements are produced by using syntax-based and/or semantic-based techniques. This allowed improving the quality of the mappings and solving the nominal conflicts.

The validation of the approach was performed by using two data sources of the health care domain. The obtained results are encouraging for what concerns the defined approach, even if the approach does not solve problems that depend on the quality of the data sources to be acquired. In particular, the quality of the constructed virtual view strongly depends on the quality of local schemas. Therefore, if a database to be considered is not normalized, it may contain redundancy and inconsistency that will be reflected in the new virtual schema. The only solution to this problem is a redesigning intervention of the local database.

In the future, further experimentation will be executed for validating the proposed approach and establishing the ranges of the threshold values assuring a good quality of the mappings.

REFERENCES

Lee, L., and Ling, W., 2003. A Methodology for Structural Conflict Resolution in the Integration of Entity-Relationship Schemas. *Knowledge and Information Systems*, Vol.5, No. 2, Springer-Verlag London Ltd.

Denivaldo, L., Hammoudi, S., de Souza, J, and Bontempo, A., 2006. Metamodel Matching: Experiments and Comparison. *Proceedings of the International Conference on Software Engineering Advances* (ICSEA'06), IEEE Computer Society, 2006.

Rahm, E., and Bernstein, P.A., 2001. A survey of approaches to automatic schema matching. *The International Journal on Very Large Data Bases*. Springer-Verlag.

Jayant, M., Bernstein, P. A., and Rahm, E., 2001. Generic Schema Matching with Cupid, *International Conference on Very Large Data Base*, Morgan Kaufmann Publishers.

Cohen, W. W., Ravikumar, P., and Fienberg, S. E., 2003. A Comparison of String Distance Metrics for Name-Matching Tasks, *Workshop on Information Integration on the Web*, American Association for Artificial Intelligence.

Giunchiglia, F., and Shvaiko Pavel, 2003. Semantic Matching. *The Knowledge Engineering Review journal*.

Giunchiglia, F., and Yatskevich, M., 2004. *Element* Level Semantic Matching, *Meaning Coordination and Negotiation workshop*.

Giunchiglia, F., Shvaiko, P., and Yatskevich M., 2004. S-Match: an Algorithm and an Implementation of Semantic Matching. *European Semantic Web Symposium*. Lecture Notes in Computer Science.

Miller, G., WordNet - About WordNet. Princeton University. [http://wordnet.princeton.edu].

AnHai,D., Madhavan, J., Domingos, P., Halevy, A., 2003. Ontology Matching: A Machine Learning Approach. *Handbook on Ontologies in Information Systems*

Huang J., Laura, R., Gutiérrez, Z., Mendoza García, B., Huhns M. N., 2005. A Schema-Based Approach Combined with Inter-Ontology Reasoning to Construct Consensus Ontologies. *AAAI Workshop on Contexts and Ontologies: Theory, Practice and Applications*. American Association for Artificial Intelligence.

Fridman Noy, N., and Musen, M. A, 2000. Algorithm and Tool for Automated Ontology Merging and Alignment. American Association for Artificial Intelligence.

Ursino, D., 2003. *Extraction and Exploitation of Intentional Knowledge from Heterogeneous Information Sources.* Springer Verlag.

Bergamaschi, S., 1997. *Un Approccio Intelligente all'Integrazione di Sorgenti Eterogenee di Informazione.*

Fong, J., Pang, F., Wong, D., and Fong, A., 2006. Schema Integration For Object-Relational Databases With Data Verification.

Chiticariu, L., Kolaitis, P. G. and Popa, L., 2008. Interactive Generation of Integrated Schemas. *SIGMOD'08.* ACM Press.

Hyunjang, K., Myunggwon, H., and Pankoo, K., 2005. A New Methodology for Merging the Heterogeneous Domain Ontologies based on the WordNet. *International Conference on Next Generation Web Services Practices*. IEEE Computer Society.

Pedersen, T., and Patwardhan, S., 2004. *WordNet: Similarity - Measuring the Relatedness of Concepts*.

Pattwardhan, S., Banerjee, S., and Pedersen, T., 2003. *Using Measures of Semantic Relatedness for Word Sense Disambiguation*.

Tan, P.N., Steinbach, M., and Kumar, V., 2005. *Introduction to Data Mining*. ISBN 0-321-32136-7.