DOI 10.1007/s10115-003-0095-8

Knowledge and Information Systems (2004) 6: 42–82

Knowledge and Information Systems

An Approach for Deriving a Global Representation of Data Sources Having Different Formats and Structures[1]

Domenico Rosaci, Giorgio Terracina and Domenico Ursino

DIMET, Università Mediterranea di Reggio Calabria, Reggio Calabria, Italy

Abstract. In this paper we propose an approach for deriving a global representation of data sources having different formats and structures. The proposed approach is based on the exploitation of a particular conceptual model for both uniformly representing such data sources and reconstructing both their intra-source and their inter-source semantics. Along with the global representation, our approach returns two support structures which improve the access transparency to stored information, namely, a set of mappings, encoding the transformations carried out during the construction of the global representation, and a set of views, allowing to obtain instances of the concepts of the global representation from instances of the concepts of the input data sources. The paper also describes a prototype which implements the proposed approach.

Keywords: Intensional information source integration; Inter-source properties; Metadata; Structured and semi-structured information sources

1. Introduction

1.1. Motivations

The enormous development of the Internet in general, and of the Web in particular, has led to great challenges in the fields of information management and exploitation. Some system architectures have been proposed in the past to allow different heterogeneous

[1] A preliminary and partial version of this paper appears under the title 'A semi-automatic technique for constructing a global representation of information sources having different formats and structure' in the Proceedings of the Conference 'Database and Expert System Applications (DEXA 2001)', Munich, September 2001.

Received 18 Jul 2001
Revised 25 Mar 2002
Accepted 2 Aug 2002

data sources to cooperate (Wiederhold, 1992; Levy et al., 1996; Garcia-Molina et al., 1997; Flesca et al., 1998). The heterogeneity of the involved data sources concerned their data models, their query and manipulation languages and their management systems. However, such architectures handled just structured data sources; i.e., for a given data source, all instances relative to the same concept had the same structure.

Nowadays existing data sources are not only heterogeneous in their data models, query and manipulation languages and management systems, but they also show quite different data representation formats and structure degrees, some of them being well structured (e.g., relational databases), others being semi-structured (e.g., XML documents) and others being unstructured (e.g., texts, images and sounds).

In addition, the development of computer networks increased the need for some kind of interaction between different data sources. In order to make this task easier, the necessity arises of guaranteeing access transparency to stored data; in its turn, this requires the definition of a global representation of all involved data sources.

Since the number and the complexity of the involved data sources may be significant, any proposal for obtaining the global representation and the support structures should be (almost) automatic. In addition, for obtaining high-quality results, it must take into account both the structure and the semantics of the involved data sources (Fankhauser et al., 1991).

1.2. General Characteristics of the Approach

In this paper we propose a semi-automatic approach for constructing a global representation of data sources having different formats and structures.

In order to handle the heterogeneity of the involved data sources, we exploit a particular conceptual model, called SDR network (Terracina and Ursino, 2000; Palopoli et al., 2001b). This has two main characteristics: (i) it is able to represent sources characterized by different data representation formats (such as XML documents, OEM graphs, E/R schemes); and (ii) it can support the process of reconstructing the semantics of involved sources.

In order to construct the global representation of a set of heterogeneous data sources, it is necessary to define their semantics (Fankhauser et al., 1991); an important support for carrying out such a task consists in the knowledge of *inter-source properties*, i.e., terminological, structural and semantic properties relating concepts belonging to different data sources (Batini and Lenzerini, 1984; Milo and Zohar, 1998; Bergamaschi et al., 1999; Mitra et al., 1999; Terracina and Ursino, 2000; Castano et al., 2001; Doan et al., 2001; Madhavan et al., 2001; Palopoli et al., 2001b). The inter-source properties our approach considers are:

- *synonymies*, indicating that two concepts, belonging to different data sources, have the same meaning;
- *homonymies*, denoting that two concepts, belonging to different data sources, have the same name but different meanings;
- *sub-source similarities*, indicating that two portions of data sources have the same meaning (since we use the SDR network as the reference conceptual model for representing data sources, in the following, we use the term *sub-net similarities* for indicating this kind of properties).

In order to derive inter-source properties, it is possible to exploit any of the techniques proposed in the literature, (e.g., Milo and Zohar, 1998; Bergamaschi et al., 1999; Mitra et al., 1999; Terracina and Ursino, 2000; Castano et al., 2001; Doan et al., 2001; Madhavan et al., 2001; Palopoli et al., 2001b). However, we have defined our own techniques for carrying out such a task; these are based on the exploitation of the properties of the SDR network conceptual model.

Observe that, differently from most of the integration methodologies proposed in the literature, the approach we propose in this paper is capable of handling not only the similarity of a node against another node (this is managed by considering synonymies) but also the similarity of a node against a group of nodes and the similarity between two groups of nodes (the latter two cases are managed by considering sub-source similarities).

Our approach stores all necessary information in an *intensional information base* (IIB), consisting of both a *metascheme* $IIB.M$ and a set of *meta-operators*. The metascheme stores all information about involved sources, their concepts, properties existing among concepts, etc. Insertions, deletions and modifications of both concepts and their properties, carried out by our approach, are realized by modifying the content of the Metascheme. The meta-operators are predefined procedures allowing either modification or querying of the metascheme. They are the only means to access it. The metascheme contains also a *set of mappings* (hereafter denoted by $IIB.M.SoM$), describing the way a concept belonging to the global SDR network has been obtained from one or more concepts belonging to input SDR networks, and a *set of views* (hereafter called $IIB.M.SoV$), allowing to obtain instances of the concepts of the global SDR network from instances of the concepts of input SDR networks.

Observe that the set of mappings and the set of views allow our approach to describe modifications performed on involved data sources during the construction of the global representation by using both input and output data sources and the set of operations which led to the output data source; in our case, maintaining both such representations is cheap, automatic and does not require a large amount of storing resources, as shown in the following. In our opinion this characteristic is particularly interesting; indeed, in the literature generally, each transformation carried out on data sources for the purpose of integration is described by providing either its input and its output or its input and the set of operations performed during the transformation (Batini et al., 1995). However, it is difficult to obtain the description of performed operations if the former representation is adopted. Similarly, it is expensive to derive the output if the latter representation is assumed.

1.3. Related Work

The problem of constructing a global representation from a set of data sources has received a great deal of attention in the literature. Initially, data sources taken into consideration were databases, i.e., structured information sources; in recent years, the enormous spread of the Internet has led to the challenge of integrating together both structured and semi-structured data sources. Here we provide an overview of the most common techniques for data source integration.

Since data source integration is a highly semantic process, first approaches proposed in the literature for carrying out such a task required a strong support of the human expert and, therefore, they were difficult to apply when the number and complexity of data sources to integrate were large (see Batini and Lenzerini, 1984, for an overview of these techniques).

In order to handle large amounts of data, a second generation of tools, requiring less intervention of the human expert, has been proposed. For carrying out their task, these tools referred to expert systems, knowledge-based systems and neural networks and compared structures by defining measures of similarity and dissimilarity (Navathe et al., 1986; Sheth et al., 1998; Hayne and Ram, 1990; Ellmer et al., 1995). Being almost automatic, these approaches were able to handle large amounts of data; however, they were based solely on the examination of data source structure and it was proved that 'purely structural considerations do not suffice to determine the semantic similarities of classes' (Fankhauser et al., 1991).

The considerations previously outlined led to a new generation of tools trying to both capture data source semantics and exploit it in the integration process. In order to define the semantics of involved data sources, these tools used (i) models (Buitelaar and Van De Riet, 1992; Johannesson, 1993), (ii) electronic dictionaries and ontologies (Collet et al., 1991; Metais et al., 1993, 1997), (iii) thesauruses (Gottard et al., 1992; Spaccapietra and Parent, 1994; Mirbel, 1995; Castano and De Antonellis, 1997; Palopoli et al., 2000). All these supports were constructed by exploiting linguistic tools.

All previously described approaches have been conceived to operate upon databases. The more recent widespread diffusion of the Internet made it necessary to develop tools for integrating both structured and semi-structured data sources. As a consequence, in recent years, a fourth generation of integration tools has been proposed. Generally, the approaches of the fourth generation extend to semi-structured data the methodologies previously proposed for databases. These integration tools are usually embedded in more complex systems managing the interoperability and the cooperation of data sources characterized by diverse data representation formats. Some of them are described below, as far as their integration approaches are concerned.

MOMIS (Bergamaschi et al., 2001). MOMIS carries out a semantic approach to data source integration based on an intensional study of involved sources. In order to realize the integration task, MOMIS constructs a common thesaurus which plays the role of a shared ontology for data sources taken into consideration. The constructed structure is exploited for determining the affinity degree associated with pairs of concepts belonging to different data sources. The integration is then realized by means of a cluster procedure which uses derived affinity degrees for determining groups of similar concepts. The result of the integration procedure is a global flat scheme representing all involved data sources.

TSIMMIS (Garcia-Molina et al., 1997). TSIMMIS exploits the self-describing Object Exchange Model (OEM) to represent data sources into consideration. The semantic knowledge is effectively encoded as a set of rules in the Mediator Specification Language (MSL); this enforces source integration at the mediator level. The exploitation of OEM and MSL allows TSIMMIS to integrate heterogeneous and semi-structured data sources.

Clio (Haas et al., 1999). Clio is based on the semi-automatic creation of mappings between data represented in a given target scheme and those relative to a source scheme. In Clio the user must supply functions describing column-level value correspondences. The system then selects enough functions to cover a maximal set of columns of the

target scheme and adds join clauses to tie together sources that supply input to the functions.

LSD (Doan et al., 2001). LSD exploits machine learning techniques to match a new data source against a previously determined global scheme. In particular, sources which LSD operates upon are XML DTDs. LSD exploits some base learners using different instance-level matching schemes; these are trained to assign tags of a mediated scheme to data instances of a source scheme. A meta-learner is used for combining the predictions of each of the base learners.

SKAT (Mitra et al., 1999). SKAT exploits first-order logic rules to express match and mismatch relationships as well as to derive new matches. The user initially provides application-specific match and mismatch relationships and then validates generated matches.

SIMS/Ariadne (Arens et al., 1993). SIMS is based on a particular description logic called LOOM. SIMS requires a model of both the application domain and the contents of each information source. The main focus of SIMS is on supporting user querying: queries in SIMS are written in the high-level uniform language of the application domain model. SIMS determines the information sources relative to a query by comparing the model of the application domain and the models associated with information sources.

GARLIC (Roth and Schwarz, 1997). GARLIC exploits the object-oriented language GDL for describing the local sources within a complex wrapper architecture; the global scheme is obtained by manually unifying the local sources by means of the so-called Garlic Complex Objects.

DLR (Calvanese et al., 1998). This integration approach is based on a particular description logic called DLR. In DLR, a data integration system is modeled at two different levels, namely, the *conceptual level*, containing a conceptual representation of the data residing in each source, and the *logic level*, containing a representation in terms of a logical data model of both the sources and the answers to queries posed to the integration system. Integrating heterogeneous data sources consists in providing a uniform access to the sources in terms of the common representation defined by the conceptual level. Queries are formulated at the conceptual level and it is necessary to specify how the source relations at the logical level relate to the elements of the conceptual level. This mapping is specified by associating with each source relation a view over the conceptual level, thus following the local-as-view approach. Such a view is expressed as a non-recursive Datalog query in which the predicates in the atoms are concepts, DLR relationships, attributes and domains of the conceptual level.

CUPID (Madhavan et al., 2001) has been conceived as an approach for solving the general problem of Scheme Match which has scheme integration among all possible applications. The CUPID approach is innovative in that it exploits an integrated use of linguistic and structural matching, context-dependent matching of shared types, and a bias towards leaf structure where much of the schema content resides.

1.4. Possible Applications

The construction of a global representation of a group of related data sources has many applications (see Rahm and Bernstein, 2001). Here we provide a brief description of some of them:

Intensional and extensional integration of data sources. These two forms of integration consider two different, yet complementary, aspects of the problem. In more detail, *intensional integration* concerns the activity of combining *schemes* of involved data sources for obtaining a global scheme representing all of them. Problems typically faced by this activity regard the derivation of synonymies, homonymies and other terminological and structural properties relating concepts represented in different schemes, and the exploitation of these properties for generating the global scheme. *Extensional integration* is the activity of producing (either virtual or materialized) data representing all instances present in the involved data sources. Problems typically considered by this operation regard (i) the recognition of all the instances belonging to different data sources and representing the same real-world instance, and (ii) the management of possible conflicting values concerning the instances taken into consideration.

Our approach allows exactly to perform the intensional integration of a group of data sources. Once this activity has been carried out, it is possible to perform their extensional integration. These two forms of integration are generally studied separately in the literature. Interestingly enough, we have derived an extensional integration technique which is related to the intensional integration approach described in this paper; we cannot describe here all details of this approach; however, the interested reader can find them in Pontieri et al. (2002).

E-commerce. In this application case, trading partners frequently exchange messages describing business transactions. Each partner uses its own message format. In order to enable systems to exchange messages, application developers must convert them among the formats required by different partners. An important task to carry out in such a context is the reconciliation of the different message schemes. A global representation of involved message formats can play a relevant role in such an application context; indeed, it could be a 'bridge' allowing passage from one message format to another.

Semantic query processing. In this case, the user specifies a query on data sources handled by the integration task and the system tries to answer it. Our approach makes user querying particularly easy; indeed, the user poses her/his query on the concepts relative to the global representation; the translation of the query from the global representation to the actual data sources can be carried out directly and automatically by exploiting the *set of views* returned by our approach.

Data and web warehouses. A data or web warehouse is a decision support system obtained from a set of information sources (Bernstein and Rahm, 2000; Rahm and Bernstein, 2001). A key operation for constructing a data warehouse consists in populating it with data stored in the corresponding data sources. In order to carry out such a task, involved data must be reconciled. A global representation might play a key role in such a context; indeed, a three-level data warehouse architecture has been defined,

which exploits the global representation of the involved data sources for constructing the reconciled data level of such an architecture (Palopoli et al., 2001b).

The construction of an integrated representation of a set of data sources is a classical problem in information systems research and many other applications have been proposed which could benefit from such an integrated representation. We have sketched here only some of them to provide an idea of their variety and relevance.

2. The SDR Network Conceptual Model

The SDR network (Palopoli et al., 2001b; Terracina and Ursino, 2000) is a conceptual model for describing data sources that allows to uniformly model most existing data representation formats as well as to derive and represent both their intra-source and their inter-source semantics (see below). An SDR network $Net(DS)$, representing a data source DS, is a rooted labeled graph:

$$Net(DS) = \langle NS(DS), AS(DS) \rangle = \langle NS_A(DS) \cup NS_C(DS), AS(DS) \rangle$$

Here, $NS(DS)$ is a set of nodes, each representing a concept of DS. Each node is identified by the name of the concept it represents. Nodes in $NS(DS)$ are subdivided into two subsets, namely, the set of *atomic nodes* $NS_A(DS)$ and the set of *complex nodes* $NS_C(DS)$. A node is atomic if it does not have outgoing arcs, complex otherwise. Since an SDR network node represents a concept, from now on we use the terms 'SDR network node' and 'concept' interchangeably.

$AS(DS)$ denotes a set of arcs; an arc represents a relationship between two concepts. More specifically, an arc A from S to T, labeled L_{ST} and denoted by $\langle S, T, L_{ST} \rangle$, indicates that the concept represented by S is semantically related to the concept denoted by T. S is called the 'source node' of A, whereas T is the 'target node' of A. At most one arc may exist from S to T.

The label L_{ST} is a pair $[d_{ST}, r_{ST}]$, where both d_{ST} and r_{ST} belong to the real interval [0, 1]. d_{ST} is called the *semantic distance coefficient*; it is used to indicate how much the concept expressed by T is semantically close to the concept expressed by S; this depends on the capability of the concept associated with T to characterize the concept associated with S. As an example, in an E/R scheme, an attribute A is semantically closer to the corresponding entity E than another entity E_1 related to E by a relationship R; analogously, in an XML document, a sub-element E_1 of an element E is closer to E than another element E_2 which E refers to by an $IDREF$ attribute. The semantic distance coefficient is obtained by considering the structural properties of the instances associated with the target node which are necessary for the definition of the source node; in particular, a coefficient is associated with each of these instances and the semantic distance coefficient is obtained as a mean of these coefficients. r_{ST} is called the *semantic relevance coefficient* and represents the fraction of *instances* of the concept denoted by S, whose complete definition requires at least one *instance* of the concept denoted by T.

An example of an SDR network is shown in Fig. 1; it corresponds to a data source describing a university. In the figure, in order to simplify the layout, a gray node having name x is used to indicate that the arc incident onto x must be considered incident onto the corresponding white node having the same name. SDR network nodes such as *Professor*, *Course* and *Student* represent the involved concepts. The arc $\langle$*Professor*, *Phone*, [0.38, 1]$\rangle$ denotes the existence of a relationship between *Professor* and *Phone*; in particular, it indicates that 100% of professors have a phone. The other arcs have an

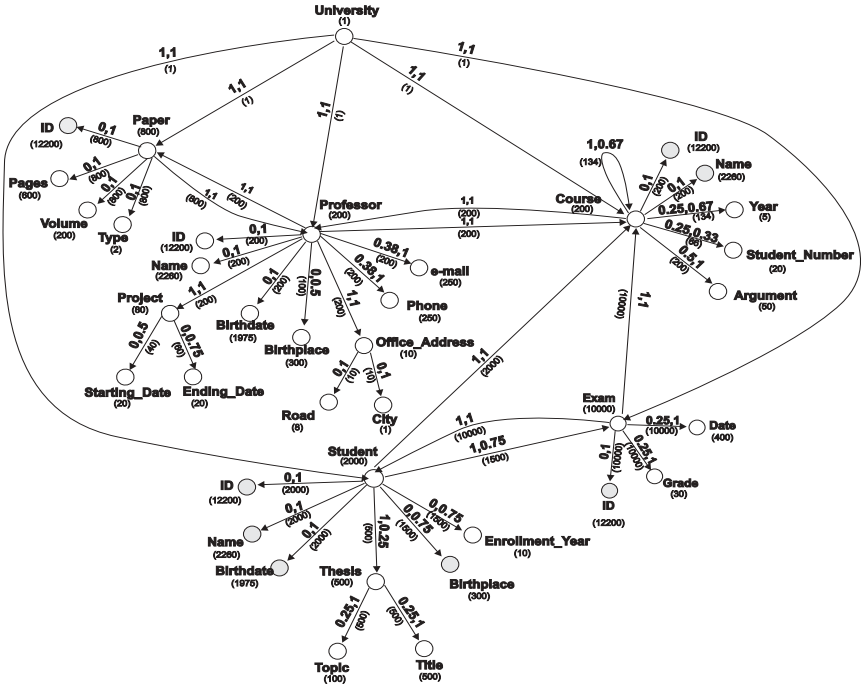

Fig. 1. The SDR network U_{S_X} representing a university.

analogous semantics. In the figure, the coefficient associated with each node indicates the number of instances of the concept the node represents; moreover, the coefficient in parentheses associated with each arc is the so-called *number of relevant instances* associated with the arc; this represents the number of instances of the source node which require at least one instance of the target node for their complete definition. This last coefficient is shown in the figure only for the sake of clarity and is not part of the model; indeed, it can be derived by multiplying the semantic relevance coefficient by the number of instances associated with the source node.

Next, we illustrate how a metrics based on the SDR network can be defined for determining the concept of neighborhood of a node which is fundamental for the computation of concept synonymies, homonymies, sub-net similarities and, overall, for obtaining a global representation of data sources. Such a metrics can be defined by means of the following definitions.

Definition 2.1. Define the *path semantic distance* of a path P in $Net(DS)$ (denoted by PSD_P) as the sum of the semantic distance coefficients associated with the arcs constituting the path.

Definition 2.2. Define the *path semantic relevance* of a path P in $Net(DS)$ (denoted by PSR_P) as the product of the semantic relevance coefficients associated with the arcs constituting the path.

Definition 2.3. Define a D_Path_n as a path P in $Net(DS)$ such that $n \leq PSD_P < n+1$.

Definition 2.4. The *CD_Shortest_Path* (Conditional D_Shortest_Path) between two nodes N and N' in $Net(DS)$ and including an arc A (denoted by $\lfloor N, N' \rfloor_A$) is the path having the minimum path semantic distance among those connecting N and N' and including A. If more than one path exists having the same minimum path semantic distance, one of those having the maximum path semantic relevance is chosen.

Definition 2.5. Given a data source DS and the corresponding SDR network $Net(DS)$, the *i_th neighborhood* of a node $x \in Net(DS)$ is defined as:

$$nbh(x, i) = \{A | A \in AS(DS),$$
$$A = \langle z, y, l_{zy} \rangle, \lfloor x, y \rfloor_A \; is \; a \; D\_Path_i, x \neq y\} \;\; i \geq 0$$

Thus, an arc $A = \langle z, y, l_{zy} \rangle$ belongs to $nbh(x, i)$ if there exists a CD_Shortest_Path from x to y, including $\langle z, y, l_{zy} \rangle$, which is a D_Path_i; note that, as such, $A \notin nbh(x, j)$, $j < i$. Finally, it is worth pointing out that x may coincide with z.

An example can help in understanding the concept of neighborhood of a node in an SDR network.

Example 2.1. Consider the node *Professor* of the SDR network illustrated in Fig. 1 (we call this network U_{X_SDR} in the rest of the paper). The neighborhoods associated with this node are the following:

$$nbh(Professor, 0) = \{\langle Professor, ID, [0, 1] \rangle, \langle Professor, Name, [0, 1] \rangle,$$
$$\langle Professor, Birthdate, [0, 1] \rangle, \langle Professor, Birthplace, [0, 0.5] \rangle\}$$

For instance, the first arc belongs to $nbh(Professor, 0)$ because $Professor \neq ID$ and $\lfloor Professor, ID \rfloor_{\langle Professor, ID, [0,1] \rangle}$ is a D_Path_0.

$$nbh(Professor, 1) = \{\langle Professor, Phone, [0.38, 1] \rangle, \langle Professor, e\text{-}mail, [0.38, 1] \rangle,$$
$$\langle Professor, Office\_Address, [1, 1] \rangle, \langle Office\_Address, Road, [0, 1] \rangle,$$
$$\langle Office\_Address, City, [0, 1] \rangle, \langle Professor, Paper, [1, 1] \rangle, \langle Paper, ID, [0, 1] \rangle,$$
$$\langle Paper, Pages, [0, 1] \rangle, \langle Paper, Volume, [0, 1] \rangle, \langle Paper, Type, [0, 1] \rangle,$$
$$\langle Paper, Project, [1, 1] \rangle,$$
$$\langle Project, Starting\_Date, [0, 0.5] \rangle, \langle Project, Ending\_Date, [0, 0.75] \rangle,$$
$$\langle Professor, Course, [1, 1] \rangle,$$
$$\langle Course, ID, [0, 1] \rangle, \langle Course, Name, [0, 1] \rangle\}$$

For instance, $A = \langle Project, Starting\_Date, [0, 0.5] \rangle$ belongs to $nbh(Professor, 1)$ because $\lfloor Professor, Starting\_Date \rfloor_A$ is a D_Path_1 and $Professor \neq Starting\_Date$. In a similar fashion, it is possible to derive all the other neighborhoods relative to U_{X_SDR}.

Note that basically any data source can be represented as a set of concepts and a set of relationships among concepts. However, data sources presently available are generally characterized by a large variety of heterogeneities ranging from their syntax to their semantics and their representation formats; finally, some of them are also multimedia. The SDR network is capable of handling the first three of these heterogeneities but it has not been conceived for managing multimedia information. Our model supports the management of syntactic and semantic heterogeneities in that it allows the extraction of inter-source properties (see Section 3).

As for data representation format heterogeneity, we observe that, presently, some data sources are completely structured (such as relational databases), while others are semi-structured (such as OEM graphs and XML documents); finally, others are unstructured (such as flat files or record archives). The SDR network model is capable

of handling different structure degrees by means of semantic distance and semantic relevance coefficients. In order to show how sources with different data representation formats can be translated into SDR networks, in Section 10 we illustrate translation rules from XML documents and E/R schemes to SDR network. We have also derived translation rules from OEM graphs to SDR networks (Terracina and Ursino, 2000). Observe that most of the other existing data representation formats (such as relational databases, record archives and HTML files) could be translated into either E/R schemes or XML documents and, consequently, into SDR networks.

3. An Overview of Inter-source Property Derivation

3.1. Derivation of Synonymies and Homonymies Between Concepts

The proposed technique for extracting synonymies and homonymies between concepts represented in two data sources DS_1 and DS_2, possibly having different data representation formats, receives both involved data sources and some lexical properties stating synonymies between names and stored in a lexical synonymy property dictionary (LSPD). Lexical synonymies are represented as triplets of the form $\langle A, B, f \rangle$, where A and B are concept names and $f \in [0, 1]$ indicates the plausibility of the property. Lexical synonymies can be automatically derived either from a standard thesaurus, such as WordNet (Miller, 1995), or by a specific tool, such as MindNet (Richardson et al., 1998). In order to obtain more refined results, it is possible to require the intervention of human domain experts. The plausibility coefficients of lexical synonymies is set as follows: for those cases where a plausibility value is not provided along with the similarity, the plausibility coefficient is set to 1; otherwise, the provided plausibility value is used. In any case the support of a human domain expert can be requested to validate and, possibly, modify the plausibility coefficients. We have carried out a study on the dependence of obtained results on possible LSPD errors; the conclusions we have drawn are described in Section 7.1.

In the following we suppose that the SDR networks $Net(DS_1)$ and $Net(DS_2)$, associated with DS_1 and DS_2, have been constructed and that actually $Net(DS_1)$ and $Net(DS_2)$, along with the LSPD, are provided as input to our technique.

Since each SDR network node is associated with one concept of the corresponding information source and vice versa, in the following we refer to the extraction of synonymies and homonymies between concepts also as the extraction of synonymies and homonymies between SDR network nodes. Moreover, since the generic concept and the corresponding SDR network node have the same name, we use this name to refer to both of them interchangeably.

The technique consists of two phases. The first one, for each pair of nodes $N_l \in Net(DS_1)$ and $N_m \in Net(DS_2)$, derives the so-called basic similarity between N_l and N_m. Basic similarities are rough properties taking into account only lexical similarities and the nearest neighborhoods of involved nodes; these properties are exploited as the starting point for deriving the real similarities. Basic similarities are represented as triplets of the form $\langle N_l, N_m, f_{lm} \rangle$, where N_l and N_m are the nodes under consideration and f_{lm} is a coefficient, in the real interval [0, 1], denoting the plausibility of the property; all basic similarities are stored in a *basic similarity dictionary* (BSD).

The second phase takes the BSD derived during the first phase as input and detects synonymies and homonymies between concepts of the data sources under consideration. First, the similarity degree associated with each tuple $\langle N_l, N_m, f_{lm} \rangle \in$ BSD is refined.

Then, the set of significant synonymies (respectively, homonymies) is constructed by selecting those pairs of nodes whose similarity degree is greater (respectively, smaller) than a certain, dynamically computed threshold th_{Syn} (respectively, th_{Hom}).

In order to refine the similarity coefficient associated with a tuple $\langle N_l, N_m, f_{lm}\rangle \in$ BSD, the technique analyzes both N_l and N_m and their neighborhoods $nbh(N_l, i)$ and $nbh(N_m, i)$, for each i such that $nbh(N_l, i) \neq \emptyset$ and $nbh(N_m, i) \neq \emptyset$. The influence of the similarity of neighborhoods of N_l and N_m on the similarity of N_l and N_m must be inversely proportional to their distance; in order to obtain this, a monotone decreasing weighting succession $\{p(i)\}$ is associated with the neighborhoods of N_l and N_m so that farthest neighborhoods have lightest weights.

Intuitively, the process of refining the similarity coefficient between N_l and N_m consists of the following steps:

- At step i:
  - Visiting, for each pair of nodes $N_l \in Net(DS_1)$ and $N_m \in Net(DS_2)$, $nbh(N_l, i)$ and $nbh(N_m, i)$.
  - Computing the similarity degree existing between $nbh(N_l, i)$ and $nbh(N_m, i)$ as an objective function associated with the maximum weight matching on a suitable bipartite weighed graph whose nodes correspond to the target nodes of the arcs composing $nbh(N_l, i)$ and $nbh(N_m, i)$.
- Computing the overall similarity degree of N_l and N_m as a weighed mean of similarity degrees of the various neighborhoods of N_l and N_m. Weights of the similarity degrees are the elements of the succession $\{p(i)\}$.

We are not describing in detail here the various steps of the computation of synonymies and homonymies between concepts, since this is beyond the scope of this paper. The interested reader is referred to Terracina and Ursino (2000).

3.2. Derivation of Sub-net Similarities

Let us now turn to the task of evaluating sub-net similarities. Consider a data source DS and the corresponding SDR network $Net(DS)$; the number of possible sub-nets that can be identified in $Net(DS)$ is exponential in the number of nodes of $Net(DS)$. To avoid the burden of analyzing such a huge number of sub-nets, we have defined a technique for singling out the most *promising* ones. The proposed technique receives two data sources DS_1 and DS_2 (represented by the corresponding SDR networks $Net(DS_1)$ and $Net(DS_2)$) and a dictionary SD of synonymies between nodes of $Net(DS_1)$ and $Net(DS_2)$. SD can be obtained by applying the technique presented in the previous section.

The technique derives the most promising pairs of sub-nets according to the following rules:

- It considers those pairs of sub-nets $[SN_i, SN_j]$ such that $SN_i \in Net(DS_1)$ is a rooted sub-net having a node N_i as root, $SN_j \in Net(DS_2)$ is a rooted sub-net having a node N_j as root, and N_i and N_j are synonyms.
- In order to select only the most promising pairs of sub-nets, having the synonym nodes N_i and N_j as roots, the technique computes the maximum weight matching on some suitable bipartite graphs obtained from the target nodes of the arcs forming the neighborhoods of N_i and N_j. In particular, given a pair of synonym nodes N_i and

N_j, the technique derives a *promising pair of sub-nets* $[SS_{i_k}, SS_{j_k}]$ for each k such that both $nbh(N_i, k)$ and $nbh(N_j, k)$ are not empty. SS_{i_k} and SS_{j_k} are constructed by determining the *promising pairs of arcs* $[A_{i_k}, A_{j_k}]$ such that $A_{i_k} \in nbh(N_i, l)$, $A_{j_k} \in nbh(N_j, l)$, for each l belonging to the integer interval $[0, k]$.

A *pair of arcs* $[A_{i_k}, A_{j_k}]$ is considered *promising* if (i) an edge between the target nodes T_{i_k} of A_{ik} and T_{j_k} of A_{jk} is present in the maximum weight matching computed on a suitable bipartite graph constructed from the target nodes of the arcs of $nbh(N_i, l)$ and $nbh(N_j, l)$, for some l belonging to the integer interval $[0, k]$; (ii) the similarity degree of T_{i_k} and T_{j_k} is greater than a certain given threshold.

The rationale underlying this technique is that of constructing promising pairs of sub-nets such that each pair is composed of the maximum possible number of pairs of concepts whose synonymy has been already stated. In this way it is probable that the overall similarity degree, resulting for each promising pair of sub-nets, will be high.

The second step of the technique for deriving sub-net similarities consists in computing the similarity degree associated with each pair of promising sub-nets; this is determined by computing the objective function associated with the maximum weight matching defined on a suitable bipartite graph, constructed from the nodes composing the sub-nets of the pair. The exploitation of the maximum weight matching as the main step for the computation of the similarity between two sub-nets $SN_i \in DS_1$ and $SN_j \in DS_2$ is justified by observing that SN_i (respectively, SN_j) can be considered similar to SN_j (respectively, SN_i) only if it is possible to determine a set of nodes belonging to SN_i (respectively, SN_j), each of which being a synonym with one of the nodes of SN_j (respectively, SN_i). The maximum weight matching is exploited for selecting this set.

The final step of the sub-net similarity derivation consists in filtering out those pairs of sub-nets having a similarity degree less than a certain, dynamically computed, threshold th_{Sim}.

Observe that the technique proposed here for deriving sub-net similarities is based on the same guidelines as the technique for detecting synonymies and homonymies described in the previous section. This is particularly interesting because, on the whole, we propose a *unified*, semi-automatic approach for deriving concept synonymies and homonymies, as well as sub-net similarities, relative to information sources having different data representation formats.

We do not provide all technical details for deriving sub-net similarities since this is beyond the scope of this paper. However, the interested reader can find them in Rosaci et al. (2001).

4. Support Intensional Information Base

Our approach exploits an intensional information base (IIB) as a support for storing information about input SDR networks, the global SDR network and the set of transformations carried out on input SDR networks for obtaining the global one. In particular, *IIB* includes a *metascheme* M, storing the information relative to involved sources, their concepts and inter-source properties among concepts, and a set of *meta-operators*, for modifying and querying the metascheme. We describe in detail the metascheme in Section 4.1 and the meta-operators in Section 4.2.

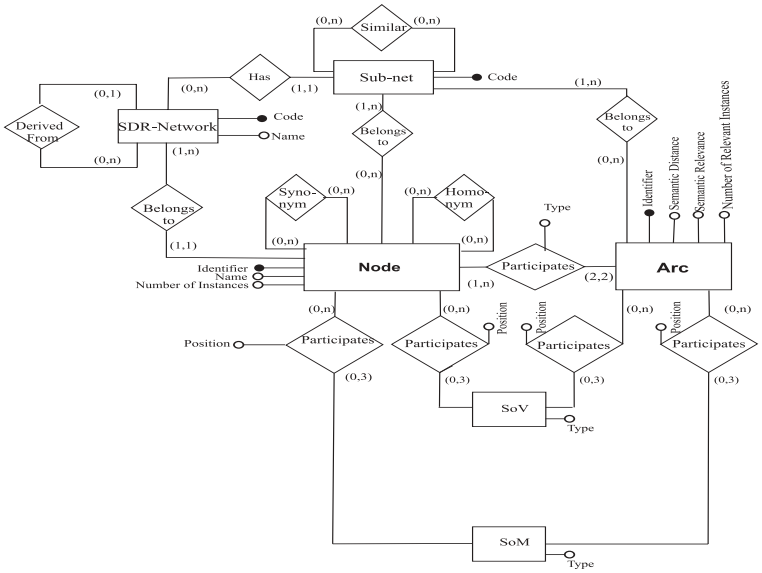

Fig. 2. The metascheme of the intensional information base.

4.1. The Metascheme

The metascheme we exploit is shown in Fig. 2. The most important entities in the metascheme are the entity *Node* and the entity *Arc*. The relationship *Participates* represents the participation of a node in an arc; its attribute *Type* indicates whether the node is the source or the target node of the arc.

A node can be a synonym with other nodes; in the Metascheme, synonymies (respectively, homonymies) are stored in the relationship *Synonym* (respectively, *Homonym*); this relationship is used for representing the synonymy dictionary SD (respectively, the homonymy dictionary HD); each tuple of SD (respectively, HD) has the form $\langle N_1, N_2 \rangle$, where N_1 and N_2 are the synonym nodes (respectively, the homonym nodes).

An SDR network can be derived from more SDR networks (this happens when it is the global SDR network, obtained by integrating the SDR networks provided as input). A sub-net is a portion of an SDR network and consists of a set of nodes and a set of arcs. A sub-net can be similar to one or more sub-nets. This situation is represented by the relationship *Similar*; this relationship is used for representing the sub-net similarity dictionary *SSD*, whose tuples have the form $\langle S_1, S_2 \rangle$, where S_1 and S_2 are the involved sub-nets.

The entity *SoM* stores the set of mappings; it can be looked at as storing the way either a node or an arc of a global SDR network has been obtained from nodes or arcs of other SDR networks by the integration algorithm. *SoM* includes an entry for each creation or modification of either a node or an arc carried out by the algorithm for constructing the global representation. Tuples of *SoM* are of the form:

- $\langle N_p, N_q, N_{pq}, NodeMerge \rangle$, indicating that the nodes N_p and N_q are merged into the node N_{pq};
- $\langle N_p, -, N_q, NodeRename \rangle$, indicating that the node N_p is renamed and transformed into the node N_q;

- $\langle R_1, R_2, R, RootCreate\rangle$, indicating that a node R is added in the same SDR network which R_1 and R_2 belong to. Here, R_1 and R_2 are the roots of two sub-nets and R becomes the 'global' root of those two sub-nets (see below);
- $\langle A_p, A_q, A_{pq}, ArcMerge\rangle$, indicating that arcs A_p and A_q are merged into the arc A_{pq};
- $\langle A_p, -, A_q, ArcChangeCoeff\rangle$, indicating that the arc A_q substitutes A_p in the corresponding SDR network. A_q has the same source and target nodes as A_p but can differ from it for the semantic distance and the semantic relevance coefficients.

The attribute *Type* of the entity *SoM* specifies the kind of the mapping. Either a node or an arc can participate in one or more tuples of *SoM*; in each tuple it participates in, it can be the first, the second or the third component of the tuple. The relationship *Participates* and its attribute *Position* store this information.

The entity *SoV* encodes the set of views. It stores a tuple for each node or arc creation or modification carried out during the integration process (and, therefore, a tuple for each tuple of *SoM*). Each view allows to obtain instances of either a node or an arc from instances of nodes or arcs it derives from. Views are defined using a 'template' language, independent from conceptual and logic scheme models, whose basic operators are parametric procedures that, once instantiated and translated into procedures valid for the management system of the data source storing data which they operates upon, compute derived data instances from input data instances. In other words, each view is an instance of one among a set of parametric views expressed using a meta-language, and is obtained (i) by substituting formal parameters with actual ones, according to *SoM* entries; (ii) by translating the obtained view from the original meta-language, which it was expressed in, into the language of the management system storing the information which the view operates upon. The set of parametric views are the following:

- $D\_NodeMerge(N_p, N_q, N_{pq})$: it is associated with merging nodes N_p and N_q into the node N_{pq} and derives instances of N_{pq} from instances of both N_p and N_q;
- $D\_NodeRename(N_p, N_q)$: it is associated with renaming the node N_p into the node N_q and derives instances of N_q from instances of N_p;
- $D\_RootCreate(R_1, R_2, R)$: it is associated with creating a root R for R_1 and R_2 (see the corresponding *SoM* entry); it derives the instances of R from those of R_1 and R_2;
- $D\_ArcMerge(A_p, A_q, A_{pq})$: it is associated with the merge of the arcs A_p and A_q into the arc A_{pq}; it allows derivation of instances of A_{pq} from instances of both A_p and A_q;
- $D\_ArcChangeCoeff(A_p, A_q)$: it is associated with the *SoM* tuple $\langle A_p, -, A_q,$ *ArcChangeCoeff*$\rangle$ and returns the instances of the arc A_q from the instances of the arc A_p.

4.2. The Meta-operators

The intensional information base is provided with a set of meta-operators; they can be classified into *meta-procedures*, that allow manipulation of the information stored in the metascheme, and *meta-functions*, that can be used for querying the metascheme. The meta-procedures are the following:

- $Add\_Node(S, N)$, which takes an SDR network S and a node N as input and adds N to S.

- $Add\_Arc(S, A)$, which receives an SDR network S and an arc A and adds A to S.
- $Delete\_Node(N)$, which takes a node N as input and deletes it from the SDR network it belongs to. It is worth pointing out that the deletion of a node from an SDR network does not imply the removal of that node from $IIB.M$. Indeed, the deleted object remains in $IIB.M$ and the information relative to its deletion from the SDR network is added. This is done in order to guarantee the capability of reconstructing the sequence of operations performed to obtain the global SDR network.
- $Delete\_Arc(A)$, which takes an arc A as input and deletes it from the SDR network it belongs to. The deletion of an arc from an SDR network is carried out by following a procedure analogous to that defined for the deletion of a node.
- $Transfer\_Incoming\_Arcs(N_x, N_y)$, which receives two nodes N_x and N_y and transfers the incoming arcs of N_x to N_y.
- $Transfer\_Outgoing\_Arcs(N_x, N_y)$, which receives two nodes N_x and N_y and transfers the outgoing arcs of N_x to N_y.
- $Set\_Node\_Name(N_x, N_y, N_{xy})$, which takes three nodes N_x, N_y and N_{xy} as input and defines the name of N_{xy} from those of N_x and N_y. This meta-procedure can require the support of the human domain expert.
- $Set\_Node\_Inst\_Number(N, n)$, which receives a node N and an integer n and sets n to be the number of instances associated with N.
- $Set\_Source(A, N)$, which takes an arc A and a node N as input and allows N to become the source node of A.
- $Set\_Target(A, N)$, which receives an arc A and a node N and allows N to become the target node of A.
- $Set\_Distance(A, d)$, which takes an arc A and a real value d as input and sets d to be the semantic distance coefficient of A.
- $Set\_Relevance(A, r)$, which receives an arc A and a real value r and sets r to be the semantic relevance coefficient of A.
- $Set\_Relev\_Inst\_Number(A, n)$, which takes an arc A and an integer n as input and sets n to be the number of relevant instances associated with A.

The meta-functions are:

- $Get\_Node\_Inst\_Number(N) \rightarrow n$, which receives a node N and returns the number n of instances associated with N.
- $Get\_Arcs(N_S, N_T) \rightarrow AS$, which takes two nodes N_S and N_T as input and returns the set AS of arcs having N_S as the source node and N_T as the target node.
- $Get\_Source(A) \rightarrow N$, which receives an arc A and returns its source node N.
- $Get\_Target(A) \rightarrow N$, which receives an arc A and returns its target node N.
- $Get\_Distance(A) \rightarrow d$, which takes an arc A as input and yields its semantic distance coefficient d as output.
- $Get\_Relevance(A) \rightarrow r$, which receives an arc A and returns its semantic relevance coefficient r.
- $Get\_Relev\_Inst\_Number(A) \rightarrow n$, which takes an arc A as input and yields the number n of relevant instances associated with A as output.
- $Get\_Nodes(S) \rightarrow NS$, which receives an SDR network S and returns the set NS of its nodes.
- $Get\_SDR\_Root(S) \rightarrow R$, which takes an SDR network S as input and returns its root R.

- $Get\_Sub\text{-}net\_Root(SN) \rightarrow R$, which receives a sub-net SN and yields its root R as output.
- $Derived\_From(N) \rightarrow NP$, which receives a node N, obtained from a merge process, and returns the pair NP of nodes which N has been derived from.

5. Construction of the Global Representation

The construction of the global representation of a group of SDR networks is carried out by an integration algorithm. This receives a support intensional information Base IIB and two SDR networks and integrates them for obtaining a global SDR network SDR_G. During the integration process the metascheme of IIB ($IIB.M$) is modified accordingly to the performed transformations.

The algorithm first determines node synonymies, node homonymies and sub-net similarities of the SDR networks given as input; in order to obtain them, it can exploit the techniques we have described in Section 3.

Involved SDR networks are then juxtaposed for obtaining a (temporarily redundant and, possibly, ambiguous) global SDR network SDR_G. In order to normalize SDR_G, by removing its redundancies and ambiguities, several transformations must be carried out on it.

The first step of SDR_G normalization consists of deriving its root.[2] In particular, if the roots of the two SDR networks in input are synonyms, they must be merged; otherwise, a new root node is created and connected to the roots of the two SDR networks.

The second step of SDR_G normalization consists of exploiting node synonymies, node homonymies and sub-net similarities for determining which nodes of SDR_G must be assumed to coincide, to be completely distinct or to be renamed. The second step is, in its turn, composed of the following sub-steps:

- *SDR network node examination*. First the synonymy dictionary SD is taken into account; for each pair $\langle N_x, N_y \rangle$ of nodes belonging to SD, N_x and N_y must be assumed to coincide in SDR_G and, therefore, must be merged into a new node N_{xy}. Then the homonymy dictionary HD is examined; for each pair $\langle N_x, N_y \rangle$ of nodes belonging to HD, N_x and N_y must be considered distinct in SDR_G and, consequently, at least one of them must be renamed.
- *SDR network arc examination*. Merging nodes produces changes in the topology of the graph; therefore, for each pair of nodes $[N_S, N_T]$ such that N_S derives from a merge process, it must be checked if N_S is connected to N_T by two arcs having the same direction[3] and, in the affirmative case, the two arcs must be merged into a unique one. If only one arc exists from N_S to N_T, the corresponding coefficients must be updated.
- *Sub-net examination*. First the sub-net similarity dictionary SSD is considered; for each pair $\langle S_x, S_y \rangle$ of sub-nets belonging to SSD, S_x and S_y must be 'merged' (the way this is done is described in the following). The merge of sub-nets could lead to the presence of two arcs connecting the same pair of nodes; if this happens, the two arcs must be merged.

The set of transformations the algorithm carries out is stored in $IIB.M.SoM$ and the corresponding views are stored in $IIB.M.SoV$. The complete algorithm for obtaining the global representation of two SDR networks is presented in Table 1.

[2] Remember that SDR networks are rooted labeled graphs.

[3] Note that this situation could happen only if also N_T derives from a merge process.

Table 1. Algorithm for constructing the global representation of two SDR networks

Input: a pair $SP = \{SDR_1, SDR_2\}$ of SDR networks; a support intensional information base IIB;
Output: a global SDR network SDR_G; a modified intensional information base IIB;
var
$Merged$, $NSet$: a set of SDR network nodes; AS: a set of SDR network arcs;
N_{xy}, N_s, N_t, R_1, R_2: an SDR network node; S_1, S_2: a sub-net;
begin
$[IIB.M.SD, IIB.M.HD, IIB.M.SSD] := Extract\_Interesting\_Properties(SP)$;
$SDR_G := Juxtaposition(IIB, SP)$;
$R_1 := IIB.Get\_SDR\_Root(SDR_1)$;
$R_2 := IIB.Get\_SDR\_Root(SDR_2)$;
if $\langle R_1, R_2\rangle \notin IIB.M.SD$ **then** $Create\_Root(IIB, R_1, R_2, SDR_G)$;
$Merged := \emptyset$;
for each $\langle N_x, N_y\rangle \in IIB.M.SD$ **do begin**
$N_{xy} := Merge\_Nodes(IIB, N_x, N_y, SDR_G)$;
$Merged := Merged \cup \{N_{xy}\}$;
end;
for each $\langle N_x, N_y\rangle \in IIB.M.HD$ **do** $Rename\_Nodes(IIB, N_x, N_y)$;
$NSet := IIB.Get\_Nodes(SDR_G)$;
for each $N_s \in Merged$ **do**
for each $N_t \in NSet$ **such that** $N_t \neq N_s$ **do begin**
$AS := IIB.Get\_Arcs(N_s, N_t)$;
if $(AS = \{A_1, A_2\})$ **then** $Merge\_Arcs(IIB, A_1, A_2, SDR_G)$;
else if $(AS = \{A_1\})$ **then** $Update\_Coefficients(IIB, A_1, SDR_G)$;
end;
for each $\langle S_1, S_2\rangle \in IIB.M.SSD$ **such that**
$\langle IIB.Get\_Sub\text{-}net\_Root(S_1), IIB.Get\_Sub\text{-}net\_Root(S_2)\rangle \notin IIB.M.SD$ **do**
$Merge\_Sub\text{-}nets(IIB, S_1, S_2, SDR_G)$;
for each $N_s \in NSet$ **do**
for each $N_t \in NSet$ **such that** $N_t \neq N_s$ **do begin**
$AS := IIB.Get\_Arcs(N_s, N_t)$;
if $(AS = \{A_1, A_2\})$ **then** $Merge\_Arcs(IIB, A_1, A_2, SDR_G)$;
end
end

$Extract\_Interesting\_Properties$ takes a pair SP of SDR networks as input and derives the synonymy dictionary, the homonymy dictionary and the sub-net similarity dictionary of the metascheme; it implements the approaches for deriving synonymies, homonymies and sub-net similarities overviewed in Sections 3.1 and 3.2.

$Juxtaposition$ receives an intensional information base IIB and a pair SP of SDR networks and juxtaposes them for obtaining a (temporarily redundant and, possibly, ambiguous) global SDR network SDR_G; in order to carry out this task it adds the suitable entries to $IIB.M$.

$Create\_Root$ creates a root for SDR_G and links it to the roots of the two SDR networks which have been juxtaposed. $Merge\_Nodes$ (respectively, $Merge\_Arcs$, $Merge\_Sub\text{-}nets$) merges the two nodes (respectively, arcs, sub-nets) received in input for obtaining a unique node (respectively, arc, sub-net).

$Rename\_Nodes$ renames at least one of the two nodes received as input; this task is carried out by adding the suitable entries to $IIB.M$; in order to decide which node

must be renamed and in order to determine the new name of the node, the procedure might need the support of the human domain expert.

Update_Coefficients updates the semantic relevance coefficient associated with the arc received as input. Note that the semantic distance coefficient does not need to be updated since it depends on the structural characteristics of the instances associated with the target node and these have not been changed. On the contrary, the semantic relevance coefficient depends on the number of instances of the source node S of A (see Section 2), and therefore it must be updated when S is obtained from a merge process.

In the following subsections we describe in detail the procedures *Create_ Root*, *Merge_Arcs*, *Update_Coefficients* and *Merge_Sub-nets* and the function *Merge_Nodes*.

5.1. Procedure Create_Root

The procedure *Create_Root* receives a support intensional information base IIB, two nodes R_1 and R_2, which are the roots of the SDR networks to integrate, and the global SDR network SDR_G. It creates a root for SDR_G. In particular, it adds to SDR_G a node R which becomes the new root. It then determines the name of R from names of both R_1 and R_2. After this, it sets the instance number of R to 1 since the unique instance of the global representation is that relative to the composition of the instances associated with R_1 and R_2. After that, it adds arcs linking R to R_1 and R to R_2. The semantic distance coefficients of these arcs are 1 because their target nodes are complex (see Terracina and Ursino, 2000, for details). The semantic relevance coefficient of the arc connecting R to R_1 (respectively, R_2) is 1 since the unique instance of R requires the support of the instance of R_1 (respectively, R_2) to be completely defined. Finally the number of relevant instances associated with the arcs connecting R to R_1 and R to R_2 is set to 1 since the (unique) instance of R_1 (respectively, R_2) is relevant for R. An entry $\langle R_1, R_2, R, RootCreate\rangle$ is added to the set of mappings and an entry $D\_RootCreate(R_1, R_2, R)$ is added to the set of views.

5.2. Function Merge_Nodes

The function $Merge\_Nodes$ takes as input a support intensional information base IIB, two nodes N_x and N_y and a global SDR network SDR_G, and merges N_x and N_y into a node N_{xy}. It first adds a new node N_{xy} to SDR_G and then derives the name, the number of instances and the arcs of N_{xy} from the corresponding ones of both N_x and N_y. Finally, it deletes N_x and N_y and adds suitable entries to both the set of mappings and the set of views. The function *Merge_Nodes* is shown in Table 2.

5.3. Procedure Merge_Arcs

The procedure $Merge\_Arcs$ receives a support intensional information base IIB, two arcs A_1 and A_2, linking the same pair of nodes, and a global SDR network SDR_G; it merges A_1 and A_2 for obtaining a unique arc. First it adds to SDR_G a new arc A_{12}, having the same source and target nodes as A_1 and A_2; then it determines the number of relevant instances,[4] the semantic distance and the semantic relevance coefficients of

[4] Recall that the number of relevant instances associated with an arc represents the number of the source node instances requiring at least one target node instance for their complete definition.

Table 2. Function $Merge\_Nodes$

Function *Merge_Nodes*(**var** IIB: a support intensional information base; N_x, N_y: an SDR network node; SDR_G: an SDR network): an SDR network node;
var
N_{xy}: an SDR network node; Num_x, Num_y: Integer;
begin
$IIB.Add\_Node(SDR_G, N_{xy})$; $IIB.Set\_Node\_Name(N_x, N_y, N_{xy})$;
$Num_x := IIB.Get\_Node\_Inst\_Number(N_x)$; $Num_y := IIB.Get\_Node\_Inst\_Number(N_y)$;
$IIB.Set\_Node\_Inst\_Number(N_{xy}, Num_x + Num_y)$;
$IIB.Transfer\_Incoming\_Arcs(N_x, N_{xy})$; $IIB.Transfer\_Outgoing\_Arcs(N_x, N_{xy})$;
$IIB.Transfer\_Incoming\_Arcs(N_y, N_{xy})$; $IIB.Transfer\_Outgoing\_Arcs(N_y, N_{xy})$;
$IIB.Delete\_Node(N_x)$; $IIB.Delete\_Node(N_y)$;
$IIB.M.SoM := IIB.M.SoM \cup \{\langle N_x, N_y, N_{xy}, NodeMerge\rangle\}$;
$IIB.M.SoV := IIB.M.SoV \cup \{D\_NodeMerge(N_x, N_y, N_{xy})\}$;
return N_{xy};
end

Table 3. Procedure $Merge\_Arcs$

Procedure *Merge_Arcs*(**var** IIB: a support intensional information base; A_1, A_2: an SDR network arc; SDR_G: an SDR network);
var
n_1, n_2, i_S, i_1, i_2: Integer; d_1, d_2, d_{12}, r_{12}: Real;
S, T, T_1, T_2: an SDR network node; A_{12}: an SDR network arc;
begin $IIB.Add\_Arc(SDR_G, A_{12})$; $S := IIB.Get\_Source(A_1)$; $T := IIB.Get\_Target(A_1)$;
$IIB.Set\_Source(A_{12}, S)$; $IIB.Set\_Target(A_{12}, T)$;
$n_1 := IIB.Get\_Relev\_Inst\_Number(A_1)$; $n_2 := IIB.Get\_Relev\_Inst\_Number(A_2)$;
$IIB.Set\_Relev\_Inst\_Number(A_{12}, n_1 + n_2)$; $[T_1, T_2] := IIB.Derived\_From(T)$;
$i_1 := IIB.Get\_Node\_Inst\_Number(T_1)$; $i_2 := IIB.Get\_Node\_Inst\_Number(T_2)$;
$d_1 := IIB.Get\_Distance(A_1)$;
$d_2 := IIB.Get\_Distance(A_2)$;
$d_{12} := \frac{i_1 \times d_1 + i_2 \times d_2}{i_1 + i_2}$; $IIB.Set\_Distance(A_{12}, d_{12})$;
$i_S := Get\_Node\_Inst\_Number(S)$; $r_{12} := \frac{n_1 + n_2}{i_S}$; $IIB.Set\_Relevance(A_{12}, r_{12})$;
$IIB.Delete\_Arc(A_1)$; $IIB.Delete\_Arc(A_2)$;
$IIB.M.SoM := IIB.M.SoM \cup \{\langle A_1, A_2, A_{12}, ArcMerge\rangle\}$;
$IIB.M.SoV := IIB.M.SoV \cup \{D\_ArcMerge(A_1, A_2, A_{12})\}$
end

A_{12}. Finally, it deletes A_1 and A_2 and adds the suitable tuples to the set of mappings and the set of views. The procedure $Merge\_Arcs$ is illustrated in Table 3.

Formulas used for obtaining d_{12} and r_{12} are justified by the following reasoning. Suppose that nodes S_1 (respectively, T_1) and S_2 (respectively, T_2) are merged into the node S (respectively, T) and that an arc A_1 (respectively, A_2) exists from S_1 (respectively, S_2) to T_1 (respectively, T_2). Furthermore, assume that A_1 and A_2 have been merged into an arc A_{12}. In order to comprehend how the formula for computing d_{12} has been derived, recall that the semantic distance coefficient is obtained by considering the structural properties of the instances associated with the target node which are necessary for the definition of the source node; in particular, a suitable coefficient is associated with each of these instances and the semantic distance coefficient is obtained as a mean

of these coefficients. As a consequence, in the computation of d_{12}, we must take into consideration the number of instances of both T_1 (hereafter i_1) and T_2 (hereafter i_2); in particular d_{12} must be obtained as a weighed mean of the semantic distance coefficients associated with A_1 and A_2, whose weights are i_1 and i_2.

In order to understand how the formula for computing r_{12} has been derived, recall that the semantic relevance coefficient is defined as the fraction of instances of the source node requiring at least one instance of the target node for their complete definition. Therefore r_{12} is obtained as a ratio whose numerator is the number of relevant instances associated with A_{12} (obtained as the sum of the number of relevant instances associated with both A_1 and A_2) and whose denominator is the number of instances of S (obtained as the sum of the number of instances of both S_1 and S_2).

5.4. Procedure Update_Coefficients

The procedure *Update_Coefficients* receives a support intensional information base IIB, an arc A and the SDR network SDR_G which A belongs to and updates the semantic relevance coefficient associated with A. In order to carry out its task, it creates a new arc A_N which substitutes A in the SDR network. A_N has the same source and target nodes as A, as well as the same semantic distance coefficient and the same number of relevant instances. The new semantic relevance coefficient is obtained as a ratio whose numerator is the number of relevant instances associated with A_N and whose denominator is the number of instances of S (the reasoning underlying the way to update the semantic relevance coefficient is the same as that drawn in Section 5.3). Finally, the procedure adds the suitable tuples to the set of mappings and the set of views.

5.5. Procedure Merge_Sub-nets

The procedure *Merge_Sub-nets* receives a support intensional information base IIB, two sub-nets SN_1 and SN_2, a global SDR network SDR_G and merges SN_1 and SN_2.

Recall that the technique for deriving sub-net similarities, overviewed in Section 3.2, takes into account only rooted connected sub-nets; as a consequence, the sub-nets SN_1 and SN_2, considered by $Merge\_Sub\text{-}nets$, have this form. This choice is due to the fact that a sub-net SN must be looked at as a unique concept, whose instances can be clearly distinguished, and this is possible only if SN has a root and is connected. Indeed, in a rooted sub-net, each instance of the sub-net has a corresponding instance at the root. Vice versa, this does not happen with unrooted sub-nets, in which case, due to the lack of a reference entry point, extensional data must be examined.

Observe, moreover, that $Merge\_Sub\text{-}nets$ is not activated when the roots of the sub-net to merge are synonyms because, in such a case, the two sub-nets are implicitly merged when the procedure $Merge\_Nodes$ is activated on the corresponding roots. The procedure *Merge_Sub-nets* is shown in Table 4.

6. Example

In this section we provide an example of the behavior of the proposed algorithm for constructing the global representation of two SDR networks. In particular, we consider the SDR network U_{S_X}, shown in Fig. 1, associated with the university web site whose DTD is presented in Fig. 10, and the SDR network U_{S_E}, illustrated in Fig. 3, associated with the university database whose E/R scheme is shown in Fig. 11 (see the Appendix

Table 4. Procedure *Merge_Sub-nets*

Procedure *Merge_Sub-nets*(**var** IIB: a support intensional information base; SN_1, SN_2: a sub-net; SDR_G: an SDR network);

var

R_1, R_2, R : a node; A_1, A_2: an arc;

begin

$R_1 := IIB.Get\_Sub\text{-}net\_Root(SN_1)$;

$R_2 := IIB.Get\_Sub\text{-}net\_Root(SN_2)$; $IIB.Add\_Node(SDR_G, R)$;

$IIB.Set\_Node\_Name(R_1, R_2, R)$; $n_1 := IIB.Get\_Node\_Inst\_Number(R_1)$;

$n_2 := IIB.Get\_Node\_Inst\_Number(R_2)$; $IIB.Set\_Node\_Inst\_Number(R, n_1 + n_2)$;

$IIB.Transfer\_Incoming\_Arcs(R_1, R)$; $IIB.Transfer\_Incoming\_Arcs(R_2, R)$;

$IIB.Add\_Arc(SDR_G, A_1)$; $IIB.Set\_Source(A_1, R)$; $IIB.Set\_Target(A_1, R_1)$;

$IIB.Set\_Distance(A_1, 1)$; $IIB.Set\_Relevance\left(A_1, \frac{n_1}{n_1+n_2}\right)$;

$IIB.Set\_Relev\_Inst\_Number(A_1, n_1)$;

$IIB.Add\_Arc(SDR_G, A_2)$; $IIB.Set\_Source(A_2, R)$; $IIB.Set\_Target(A_2, R_2)$;

$IIB.Set\_Distance(A_2, 1)$; $IIB.Set\_Relevance\left(A_2, \frac{n_2}{n_1+n_2}\right)$;

$IIB.Set\_Relev\_Inst\_Number(A_2, n_2)$;

$IIB.M.SoM := IIB.M.SoM \cup \{\langle R_1, R_2, R, RootCreate\rangle\}$;

$IIB.M.SoV := IIB.M.SoV \cup \{D\_RootCreate(R_1, R_2, R)\}$

end

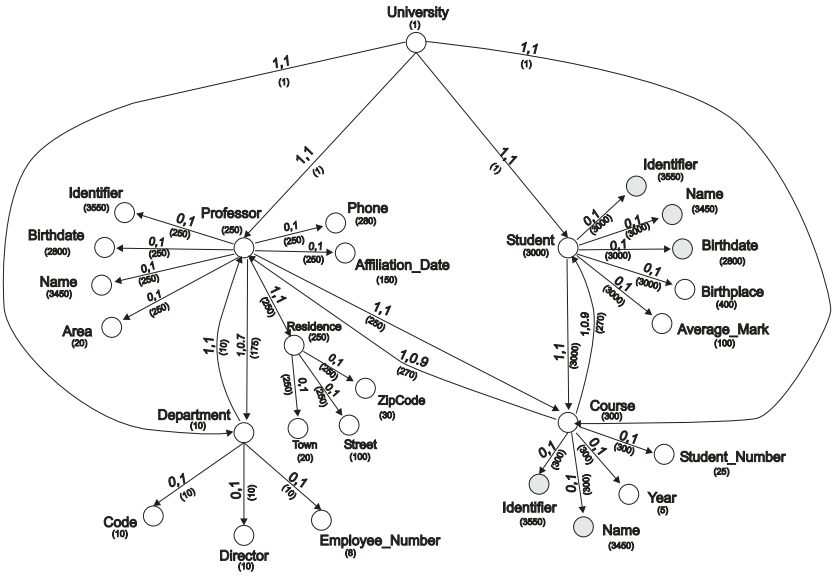

Fig. 3. The SDR network U_{S_E} representing a university.

for details about the construction of U_{S_X} and U_{S_E} from the corresponding information sources). The global SDR network the proposed algorithm derives is shown in Fig. 4. Recall that, in these figures, a gray node having name N is used to indicate that the arc incident onto N must be considered incident onto the corresponding white node having the same name.

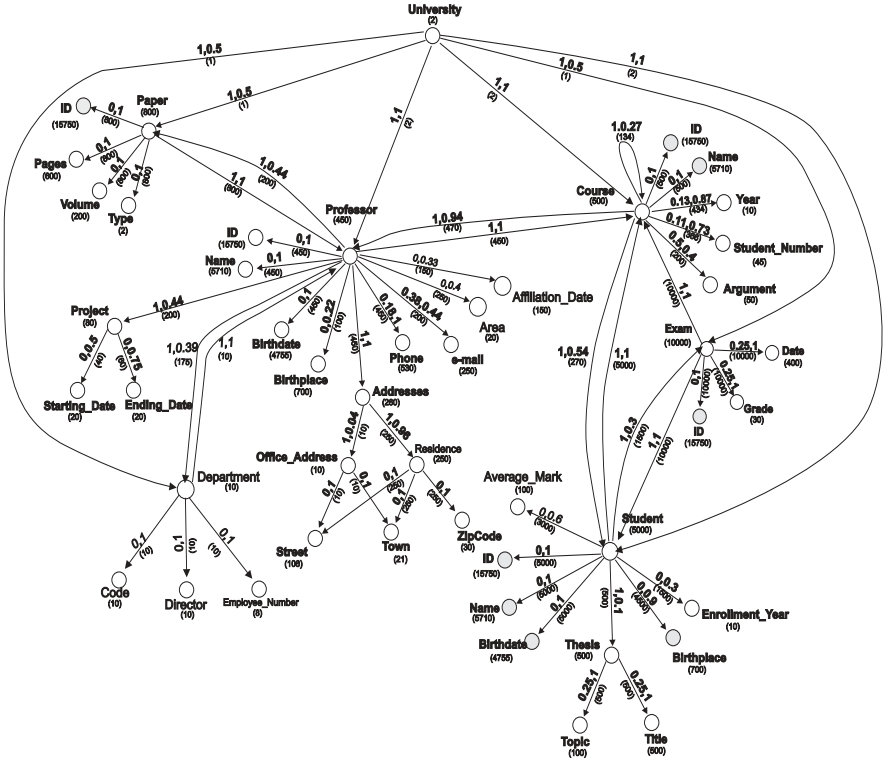

Fig. 4. The global SDR network.

The algorithm first activates the function *Extract_Interesting_Properties* on U_{S_X} and U_{S_E}; the synonymy dictionary this function derives from is shown in Table 5.[5]

In addition *Extract_Interesting_Properties* detects a similarity between the sub-nets $SN_1 = \langle Office\_Address - Road - City \rangle$ of U_{S_X} and $SN_2 = \langle Residence - Town - Street - Zip\_Code \rangle$ of U_{S_E}.

After that, the function *Juxtaposition* is activated; it juxtaposes U_{S_X} and U_{S_E} and obtains a (temporarily redundant and, possibly, ambiguous) global SDR network U_G. Then, a check is carried out to verify if the roots of U_{S_X} and U_{S_E} are synonyms or not. Since they are synonyms, the procedure $Create\_Root$ is not activated.

Then, for each pair of synonym nodes stored in the synonymy dictionary, $Merge\_Nodes$ is activated. In particular, when it is called on $Student_{[U_{G\langle U_{S_X}\rangle}]}$[6] and $Student_{[U_{G\langle U_{S_E}\rangle}]}$, a new node is added to U_G; Set_Node_Name sets its name to $Student_{[U_G]}$ whereas *Set_Node_Inst_Number* sets its number of instances to the sum of the number of instances of $Student_{[U_{G\langle U_{S_X}\rangle}]}$ (2000) and $Student_{[U_{G\langle PG\rangle}]}$ (3000). After this, arcs of $Student_{[U_{G\langle U_{S_X}\rangle}]}$ and $Student_{[U_{G\langle U_{S_E}\rangle}]}$ are transferred to $Student_{[U_G]}$. Then $Student_{[SDR_{G\langle U_{S_X}\rangle}]}$ and $Student_{[SDR_{G\langle U_{S_E}\rangle}]}$ are deleted from U_G.

[5] Here and in the following, $N_{[S]}$ indicates node N of the SDR network S.

[6] Here and in the following, $N_{[G\langle I\rangle]}$ indicates node N of the global SDR network G, which has been directly derived from the corresponding node of the input SDR network I by juxtaposition.

Table 5. The synonymy dictionary relative to the SDR networks U_{S_X} and U_{S_E}

First node	Second node
$\text{University}_{[U_{S_X}]}$	$\text{University}_{[U_{S_E}]}$
$\text{Professor}_{[U_{S_X}]}$	$\text{Professor}_{[U_{S_E}]}$
$\text{Student}_{[U_{S_X}]}$	$\text{Student}_{[U_{S_E}]}$
$\text{Course}_{[U_{S_X}]}$	$\text{Course}_{[U_{S_E}]}$
$\text{ID}_{[U_{S_X}]}$	$\text{Identifier}_{[U_{S_E}]}$
$\text{Name}_{[U_{S_X}]}$	$\text{Name}_{[U_{S_E}]}$
$\text{Birthdate}_{[U_{S_X}]}$	$\text{Birthdate}_{[U_{S_E}]}$
$\text{Birthplace}_{[U_{S_X}]}$	$\text{Birthplace}_{[U_{S_E}]}$
$\text{Year}_{[U_{S_X}]}$	$\text{Year}_{[U_{S_E}]}$
$\text{Student\_Number}_{[U_{S_X}]}$	$\text{Student\_Number}_{[U_{S_E}]}$
$\text{Road}_{[U_{S_X}]}$	$\text{Street}_{[U_{S_E}]}$
$\text{Town}_{[U_{S_X}]}$	$\text{City}_{[U_{S_E}]}$

Finally, the tuple $\langle Student_{[U_{G_{\langle U_{S_X}\rangle}}]}, Student_{[U_{G_{\langle U_{S_E}\rangle}}]}, Student_{[U_G]}, NodeMerge\rangle$ is added to the set of mappings and the tuple $D\_NodeMerge(Student_{[U_{G_{\langle U_{S_X}\rangle}}]}, Student_{[U_{G_{\langle U_{S_E}\rangle}}]}, Student_{[U_G]})$ is added to the set of views. *Merge_Nodes* works analogously for the other pairs of nodes stored in the synonymy dictionary. Since no homonymy has been found on U_{S_X} and U_{S_E}, the function *Rename_Nodes* is not activated.

After all nodes have been examined, if two arcs exist between the same pair of nodes, they must be merged; this task is carried out by activating the procedure *Merge_Arcs* on them. As an example, due to the merge of $Course_{[U_{G_{\langle U_{S_X}\rangle}}]}$ and $Course_{[U_{G_{\langle U_{S_E}\rangle}}]}$ into $Course_{[U_G]}$ and the merge of $Professor_{[U_{G_{\langle U_{S_X}\rangle}}]}$ and $Professor_{[U_{G_{\langle U_{S_E}\rangle}}]}$ into $Professor_{[U_G]}$, there are two arcs from $Course_{[U_G]}$ to $Professor_{[U_G]}$, namely $A_1 = \langle Course_{[U_G]}, Professor_{[U_G]}, [1, 1]\rangle$ and $A_2 = \langle Course_{[U_G]}, Professor_{[U_G]}, [1, 0.9]\rangle$ and the procedure $Merge\_Arcs$ must be activated on them. It adds a new arc A_{12} to U_G. The source and the target nodes of A_{12} are $Course_{[U_G]}$ and $Professor_{[U_G]}$, respectively. The number of relevant instances of A_{12} is given by the sum of the number of relevant instances of A_1 (200) and that of A_2 (270). Since the semantic distance coefficient of A_1 (respectively, A_2) is 1 (respectively, 1) and since the number of instances of $Professor_{[U_{G_{\langle U_{S_X}\rangle}}]}$ (respectively, $Professor_{[U_{G_{\langle U_{S_E}\rangle}}]}$) is 200 (respectively, 250), then the semantic distance coefficient of A_{12} is $\frac{200\times1+250\times1}{450} = 1$. Since the number of instances of $Course_{[U_G]}$ is 500, then the semantic relevance coefficient for r_{12} is $\frac{470}{500} = 0.94$. Finally, A_1 and A_2 are deleted, the tuple $\langle A_1, A_2, A_{12}, ArcMerge\rangle$ is added to the set of mappings and the tuple $D\_ArcMerge(A_1, A_2, A_{12})$ is added to the set of views. All the other pairs of arcs connecting a pair of merged nodes are handled in a similar manner.

As for the other arcs connected to a source node obtained by the merge of two nodes, the algorithm activates the procedure *Update_Coefficients* on them. One such arc is that between $Course_{[U_G]}$ and $Student_{[U_G]}$ – we call this arc A_3 in the following. In this case the procedure first determines its source ($Course_{[U_G]}$), its target ($Student_{[U_G]}$), its semantic distance coefficient (1) and its number of relevant instances (270). Then it adds a new arc A_N from $Course_{[U_G]}$ to $Student_{[U_G]}$ having the same semantic distance coefficient and the same number of relevant instances as A_3; the

semantic relevance coefficient is set to $\frac{270}{500} = 0.54$. After this, it deletes A_3 from SDR_G and adds the tuples $\langle A_3, -, A_N, ArcChangeCoeff \rangle$ to the set of mappings and $D\_ArcChangeCoeff(A_3, A_N)$ to the set of views.

After all arcs have been examined, the algorithm takes into account sub-net similarities. In particular, it must consider only those pairs of similar sub-nets whose roots are not synonyms (see Section 5.5); to this aim it activates the function *Merge_Sub-nets* on each of these pairs. In particular, when *Merge_Sub-nets* is activated on SN_1 and SN_2, it first determines the roots of SN_1 and SN_2 (i.e., $Office\_Address_{[SDR_{G_{\langle U_{S_X} \rangle}}]}$ and $Residence_{[U_{G_{\langle U_{S_E} \rangle}}]}$, respectively). After this, a root node, named *Addresses*, is added to U_G. The number of instances of $Addresses_{[U_G]}$ is $10 + 250 = 260$. Arcs incoming into $Office\_Address_{[U_{G_{\langle U_{S_X} \rangle}}]}$ and $Residence_{[U_{G_{\langle U_{S_E} \rangle}}]}$ are transferred to $Addresses_{[U_G]}$. In addition, the arc A, linking $Addresses_{[U_G]}$ and $Office\_Address_{[U_{G_{\langle U_{S_X} \rangle}}]}$, is added; its semantic distance coefficient is set to 1; its semantic relevance coefficient is set to $\frac{10}{10+250} = 0.04$ and its number of relevant instances is set to 10. Analogously, an arc A', linking $Addresses_{[U_G]}$ to $Residence_{[U_{G_{\langle U_{S_E} \rangle}}]}$, is added to U_G; the semantic distance coefficient of A' is set to 1; its semantic relevance coefficient is set to $\frac{250}{10+260} = 0.96$ whereas its number of relevant instances is set to 250.

Finally, $\langle Office\_Addresses_{[U_{G_{\langle U_{S_X} \rangle}}]}, Residence_{[U_{G_{\langle U_{S_E} \rangle}}]}, Addresses_{[U_G]}, RootCreate \rangle$ is added to the set of mappings whereas $D\_RootCreate(Office\_Address_{[U_{G_{\langle U_{S_X} \rangle}}]}, Residence_{[U_{G_{\langle U_{S_E} \rangle}}]}, Addresses_{[U_G]})$ is added to the set of views.

Observe that the algorithm returns a global SDR network that uniformly represents both U_{S_X} and U_{S_E}. It is worth pointing out that the construction of the global SDR network led us also to obtain the set of mappings and the set of views which can be exploited for improving access transparency to data sources under consideration.

7. Quality of Results

7.1. Stability of Inter-source Property Derivation Against LSPD Errors

In order to verify the stability of our techniques against errors potentially occurring in LSPD coefficients, we have carried out some sensitivity analyses. In performing such a task we have considered various cases. For each of them we have measured some parameters, namely, *maximum increment*, *mean variation* and *maximum decrement* in returned plausibility coefficients, *changes of similarity threshold values* and *differences in the set of recognized synonymies* (recall that, in our derivation approach, synonymy threshold values are dynamically computed and that all coefficients belong to the real interval [0, 1]).

As for errors on LSPD coefficients provided by the expert, to understand the influence of such errors we have considered a percentage of wrong entries equal to (i) 10%, (ii) 30%, (iii) 50% of the total. For each of these cases we have considered six situations: (a) all wrong entries are underestimated by 10%; (b) all wrong entries are overestimated by 20%; (c) all wrong entries are underestimated by 40%; (d) all wrong entries are overestimated by 40%; (e) half of the wrong entries are underestimated by 20% and half of them are overestimated by 20%; (f) half of the wrong entries are underestimated by 40% and half of them are overestimated by 40%.

Table 6. Values of quality parameters for case (i)

	(a)	(b)	(c)	(d)	(e)	(f)
Maximum increment	0.000	0.000	0.000	0.056	0.000	0.000
Mean variation	0.000	−0.008	−0.009	−0.009	−0.001	−0.009
Maximum decrement	0.000	0.006	0.000	0.012	0.000	0.006
Synonymy threshold change	0.000	−0.009	0.000	−0.013	−0.001	−0.011
Differences of recognized synonymies	None	None	None	None	None	None

Table 7. Measures of the soundness and completeness of our approach

	Soundness	Completeness
Synonymies	98%	92%
Homonymies	98%	98%
Object cluster similarities	96%	88%

Results of our experiments for cases (i)(a–f) are provided in Table 6. Due to space limitations, we cannot show here all details about the other computations. The interested reader can find them in Ursino (2000).

The examination of all such experiments has shown that our inter-source property techniques present a good stability w.r.t. the errors in providing LSPD entries. Even if errors occur when the LSPD is constructed, the changes in the obtained plausibility values yielded by our techniques are generally quite small. In addition, even if changes in the obtained plausibility values are significant (i.e., greater than 5%), recognized synonymies do not necessarily change, since our thresholds are dynamic, being computed as functions of plausibility coefficients.

7.2. Soundness and Completeness

In Table 7 we present a brief summary about the quality of results obtained by running our inter-source property derivation algorithm. In the table we measure their soundness and completeness. In particular, the *soundness* lists the percentage of properties returned by our techniques agreeing with those provided by human experts, whereas the *completeness* lists the percentage of returned properties w.r.t. the set of properties provided by the human experts. In order to verify how much the soundness and the completeness figures vary with the modifications in the LSPD precision, we conducted a further set of experiments for several data source families. In particular, we considered the following set of variations in the LSPD: (i) 10% of entries are underestimated by 20%; (ii) 10% of entries are overestimated by 40%; (iii) 10% of entries are underestimated by 60%; (iv) 30% of entries are underestimated by 20%; (v) 30% of entries are overestimated by 40%; (vi) 30% of entries are underestimated by 60%; (vii) 50% of entries are underestimated by 20%; (viii) 50% of entries are overestimated by 40%; (ix) 50% of entries are underestimated by 60%. Figure 5 shows results relative to soundness and completeness variations. Note that even relatively significant variations in the LSPD did not substantially affect either soundness or completeness. This is mainly due to the dynamic computation of the thresholds for selecting final properties, making our technique able to adaptively reconfigure its behavior against limited variations in the values of LSPD entries so that yielded results do not change much. This confirms

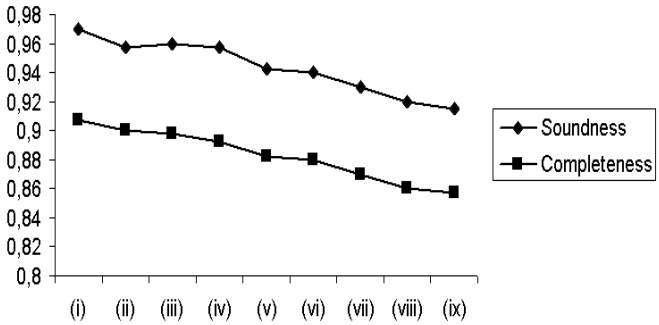

Fig. 5. Variations of soundness and completeness w.r.t. variations in the LSPD.

that the results produced by our technique are largely influenced by the structure of the objects belonging to analyzed schemes and of their dependencies, which together determine scheme semantics, rather than syntactic characteristics that are found therein (specifically, object names).

Finally, it is worth pointing out that soundness and completeness are related to dynamic thresholds. Due to space limitations, we cannot describe here the structure of such thresholds; the interested reader can find it in Palopoli et al. (2001b). However, we observe that, if we want a higher completeness, it is necessary that threshold value decreases; in this case, however, the soundness degree becomes smaller. Vice versa, if thresholds are increased, the soundness becomes higher and the completeness becomes smaller. In our opinion, our choices concerning thresholds allow us to obtain a good trade-off between soundness and completeness.

Finally, observe that the completeness associated with sub-net similarities is smaller than that relative to synonymies and homonymies. This behavior is justified by considering that the number of sub-net pairs relative to two data sources is exponential w.r.t. the number of objects represented in those sources (see Section 3.2). As a consequence, in order to make our approach feasible, we have designed a heuristics which returns a polynomial number of promising pairs; we compute the similarity degree only for these pairs. As for soundness and completeness of the integration, it is worth observing that if a similarity is not detected by our derivation technique the integration algorithm does not merge the corresponding concepts. The corresponding global scheme will be redundant in some parts but it will not present errors. Vice versa, if a returned similarity is not correct, it could produce errors in the global scheme, thus influencing the soundness of the whole approach; for this reason we require a human expert for validating obtained similarities before starting the integration activity.

Obviously, if we increase the value of the threshold, the soundness of the properties will be higher and the human expert required contribution would be smaller; however, in our opinion, this is not the right way to operate because:

- The completeness is reduced and, therefore, a greater number of existing properties would not be derived.
- It was shown that purely structural considerations are not sufficient to derive inter-source properties (Fankhauser et al., 1991); as a consequence, in their extraction, it is necessary to consider a semantic component reflecting the way the reality of interest has been represented by designers (in our approach the semantic component is

represented by the concept of neighborhood and we take it into account by computing the similarities of neighborhoods when we determine the similarity of two objects (see Section 3.1). The presence of a semantic and, in some way, subjective component in the computation of inter-source properties makes it impossible to have a formal proof of their soundness (i.e., the limit of soundness tends to 1 but it is impossible to formally prove that soundness is 1). Such reasoning is evidence that a human expert validating the properties is, in any case, necessary.

- Since human validation is required, we prefer to have a certain trade-off between soundness and completeness so that the number of properties to examine is quite low and almost all of them are correct.

7.3. Time and Space Complexity

The time complexity of our algorithm for constructing the global represen- tation of two SDR networks SDR_1 and SDR_2 is polynomial in the number of nodes Num_1 of SDR_1 and Num_2 of SDR_2.

In more detail, the function $Extract\_Interesting\_Properties$ is polynomial in Num_1 and Num_2, $Juxtaposition$ is quadratic in Num_1 and Num_2, and the merge of both synonym nodes and arcs and similar sub-sources is quadratic in Num_1 and Num_2.

The space complexity of our algorithm for constructing the global representation of two SDR networks SDR_1 and SDR_2 is quadratic in the number of nodes Num_1 of SDR_1 and Num_2 of SDR_2.

8. Comparisons Between our Approach and Related Ones

In our opinion, since the SDR network model is well suited for uniformly handling data sources with heterogeneous representation formats (see Section 2), our SDR network-based intensional integration approach appears particularly adequate as a fourth-generation integration tool. In this section we provide a complete comparation between the features of our approach and the main characteristics of the other fourth-generation integration tools described in Section 1.3.

MOMIS (Bergamaschi et al., 2001). Both the MOMIS approach and our own exploit a semantically rich conceptual model for deriving and representing the semantics of each involved data source. Both of them allow the semi-automatic extraction of inter-source properties relating concepts (or sub-schemes) of data sources at the conceptual level. Both of them allow a representation of the mapping between the data at the sources and the integrated representation. As for differences, the conceptual model exploited by MOMIS is object-oriented, whereas that used by our approach is graph-based; in addition, the approach we propose detects not only synonymies and homonymies but also sub-source similarities which MOMIS does not take into account.

TSIMMIS (Garcia-Molina et al., 1997). Both TSIMMIS and our approach allow the construction of a mediator carrying out the integration of a group of data sources being heterogeneous in their structure, semantics and data representation formats. However, some differences can be noticed between the two approaches. Indeed, TSIMMIS exploits the OEM conceptual model for representing involved sources; our approach uses the SDR network, which is capable of uniformly handling various source formats. TSIMMIS is based on a *structural* approach whereas our approach is semantic.

Clio (Haas et al., 1999). The main difference between the Clio approach and our own is that Clio handles together, via a uniform mechanism, both scheme transformations and data transformations within the integration task. This is done by exploiting object-extended SQL functionalities at both the wrappers level and the middleware level. However, users must supply the so-called *column level value correspondencies*, i.e., concept synonymies. Therefore Clio does not take into account sub-source similarities and also concept synonymies are provided manually by the expert. On the contrary, our approach autonomously derives *all* inter-source properties necessary for obtaining the global representation. Finally, data transformations are not handled by our approach but, conversely, we are able to handle a larger variety of structural and semantic heterogeneities.

LSD (Doan et al., 2001). There are several differences between the approach proposed in LSD and our own. First, it exploits machine learning techniques, whereas we use graph-based techniques to derive source semantics and inter-relationships between concepts. LSD is able to handle just XML documents, whereas we can manage sources characterized by heterogeneous data definition formats. LSD looks for mappings between source schemes and the global scheme, but does not derive the global scheme; on the contrary, we derive a global representation of input data sources, thus generating the mappings. Moreover, LSD is capable of deriving only 1–1 mappings between the elements of two XML documents and does not consider sub-source similarities possibly existing between input sources. However, mappings extracted by LSD are supported by source data analysis and, moreover, they are associated with plausibility coefficients indicating the confidence of the learner with that mapping.

SKAT (Mitra et al., 1999).The approach exploited in SKAT is similar to our own in that it is semi-automatic, exploits graphs to represent information sources and looks at both concept and sub-source similarities to derive the matchings. Moreover, it uses on-line dictionaries to determine term-based matching rules. The main differences between the approach implemented in SKAT and our own are the following. First, the result of SKAT is a unified representation of only those portions of sources found to match, whereas we obtain a uniform and global representation of all the input sources. Moreover, in SKAT, graph representation of web sources consists of one node per web page, whereas we derive *one* graph representation for *each* web page; in this way we are able to obtain more refined representations of involved sources. Finally, SKAT exploits first-order logic rules to express derived matchings. This allows SKAT to represent possibly complex relationships between concepts, but makes necessary a heavy intervention of human experts in the derivation phase of such relationships. On the contrary, we represent relationships between concepts, or group of concepts, by means of tuples, but human intervention is required only in the validation of automatically derived matchings.

SIMS/Ariadne (Arens et al., 1993). Both the Ariadne approach and that which we present in this paper allow creation of a unified view of a set of semi-structured information sources. However, Ariadne deals with HTML information sources whereas the SDR network handles a large variety of source formats. Ariadne derives only textual inter-source properties since it considers concept names, acronyms, abbreviations and phrase orderings. For deriving them it exploits information retrieval techniques. Vice versa, inter-source properties which our approach derives are both structural and semantic. Finally, the engine underlying Ariadne is based on Description Logics whereas our approach is graph centered.

GARLIC (Roth and Schwarz, 1997). The approach underlying GARLIC is quite different from our own. First, in GARLIC, the global scheme is obtained manually whereas our approach is semi-automatic. Moreover, GARLIC exploits object-oriented features for describing source data, whereas our model is graph-based. Actually, the GARLIC approach could be exploited as a complement to our own to answer queries posed on the global representation we automatically derive.

DLR (Calvanese et al., 1998). The approach of DLR and our own present some similarities in that: (i) they exploit a semantically rich representation of involved data sources through a common conceptual model which is used for deriving and representing the semantics of each involved data source; (ii) they construct an integrated and unified representation of the involved data sources which is used for querying the integration systems; (iii) they provide the mappings between the data at the sources and the integrated representation. However, some differences exist between DLR and our approach in that (i) DLR has a greater expressive power as opposed to the SDR network (except that plausibility factors are not provided) but, because of this, is more demanding from the computational complexity viewpoint; (ii) no approach is proposed by DLR for extracting inter-source properties relating either concepts or sub-schemes or data sources at the conceptual level.

CUPID (Madhavan et al., 2001). The approach underlying Cupid and our own share some peculiarities; indeed both of them (i) operate at the intensional level, (ii) exploit graph-based techniques for representing source schemes, (iii) derive inter-source properties, and (iv) exploit both structural and semantic analyses for deriving similarities. The most relevant difference between Cupid and our approach consists in their main focus; indeed, Cupid has been mainly conceived for carrying out a generic scheme match activity even if the integration task is also studied. The approach we present in this paper is mainly focused on scheme integration, and the extraction of inter-source properties is considered as a step of the integration activity. Another difference between the two approaches is due to the fact that Cupid *also* exploits linguistic considerations for deriving similarities, in addition to structure and semantics considerations; our approach, instead, integrates structure and semantic similarities with information derived from relevance which takes into account both the structure degree of the information sources and the instances associated with each concept.

9. A Prototype Implementing the Proposed Approach

In this section we describe the behavior of the prototype implementing the approach for data source integration presented in this paper.

The prototype receives an intensional information base, storing information about involved data sources and their inter-source properties. Since an intensional information base could store information about many data sources, it is necessary to allow the user to choose the data sources he/she wants to integrate. The form for carrying out this task is depicted in Fig. 6. Interestingly enough, the prototype also allows that the data sources to integrate are, in their turn, integrated data sources; in this way it is possible to handle various levels of integration.

After the user has selected the data sources to integrate, he/she is required to choose the name of the integrated SDR network.

At this point, the prototype begins to construct the integrated SDR network. The first task of this process consists of translating the data sources chosen by the user

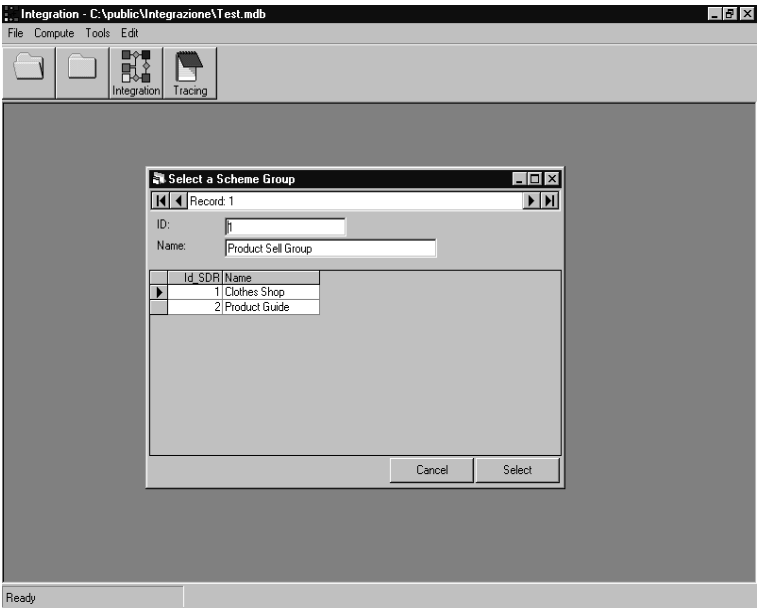

Fig. 6. Choice of the data sources to integrate.

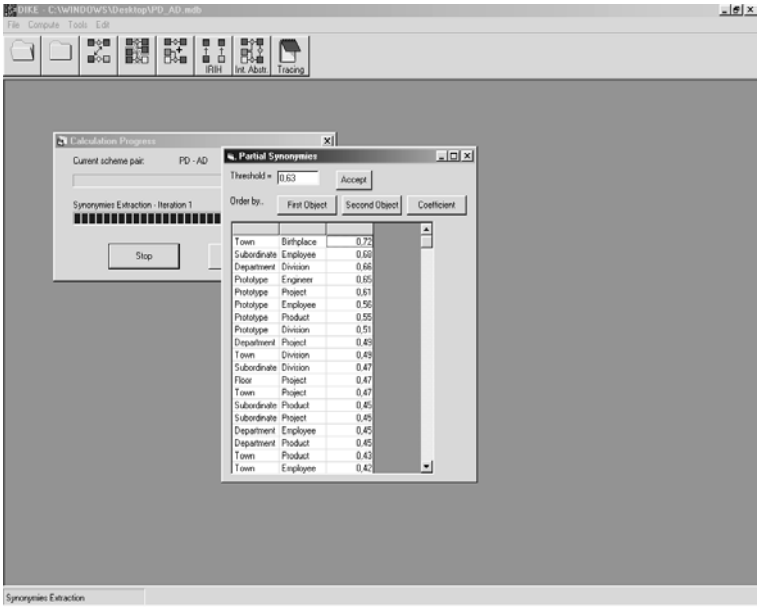

Fig. 7. Presentation of derived synonymies.

into the corresponding SDR networks; after this, synonymies, homonymies and sub-source similarities relative to these sources are computed. Figure 7 shows how derived synonymies are presented to the user. After these properties have been validated by the

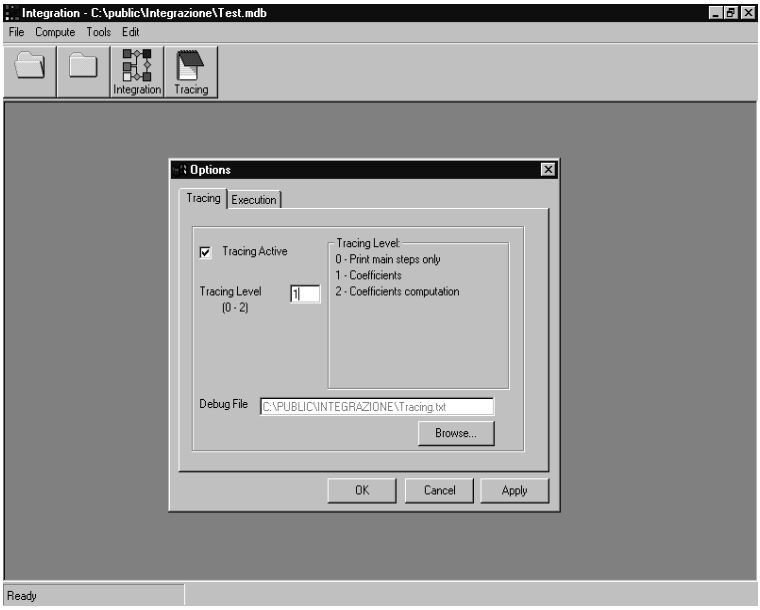

Fig. 8. Choice of the tracing level and file.

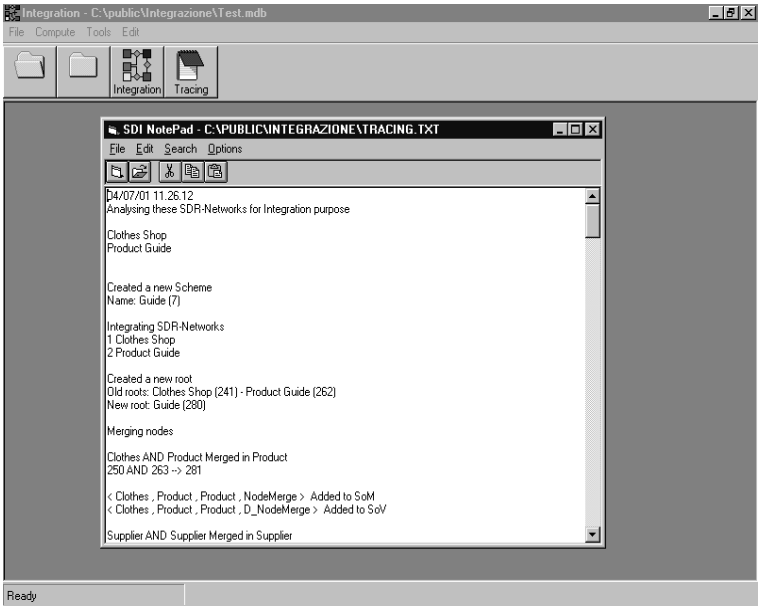

Fig. 9. Examination of the tracing file.

user, the involved SDR networks are first juxtaposed and, then, normalized. The first step of their normalization consists of deriving its root; in order to decide the name of the root, user support may be required.

After this, the prototype examines the nodes of the SDR network to integrate; in particular, it takes into account synonymies and homonymies for deciding both the nodes to merge and those to rename. The support of the user is required only when the names of the newly created nodes cannot be automatically determined.

Next, SDR network arcs are analyzed for eliminating inconsistencies; this phase is completely automatic.

Finally, sub-nets are examined for merging the similar ones. During this phase, it might be necessary to create a new sub-net root; as in the previous cases, if the name of the root cannot be automatically determined, the prototype requires user support.

The final integrated SDR network, along with the set of mappings and the set of views, are stored in the intensional information base. In addition, all this information, along with a tracing of the various operations the prototype has carried out, is stored in a suitable file. The tracing can be executed at various abstraction levels. The user can choose both the desired abstraction level and the file which the tracing must be stored on; Fig. 8 illustrates the corresponding form whereas, in Fig. 9, a fragment of the resulting tracing file is shown.

10. Conclusions

In this paper we have proposed an approach for the construction of a global representation of data sources having different data representation formats. The proposed approach exploits a conceptual model which allows uniform representation of data sources under consideration as well as reconstruction of their intra-source and inter-source semantics. It enriches the constructed global representation with two support structures allowing improvement of access transparency to stored information, namely, a set of mappings, encoding the transformations carried out during the construction of the global representation, and a set of views, allowing to obtain instances of the concepts of the global representation from instances of the concepts of the input data sources. We have also described a prototype implementing the proposed approach.

Currently, we are studying the possibility of applying the same guidelines (i.e., the exploitation of both SDR networks and the intensional information base) for defining semi-automatic approaches aimed at carrying out other kinds of transformations on data sources (e.g., the abstraction of a global data source; Palopoli et al., 2001a), the construction of a hierarchy of clusters of SDR networks allowing their categorization on the basis of their structural and semantic content.

In the future we plan to extend the integration and, more in general, the cooperation of a set of information sources taking into account not only their structural and semantic information content but also the perception the user has of the information stored in a source. As an example, consider the information stored in a site, say *Ferrari*. A Formula 1 fan is particularly interested in information about grand prix; a car collector probably examines the site for information about the last model of cars realized by Ferrari; finally, a shareholder is interested in Ferrari sales. This simple example clearly shows that the same information source could be perceived in different ways by different customers and, in our opinion, this fact heavily influences information source cooperation. As an example, if we consider the site categorization, a fan puts the site in the category *sport*, a car collector associates it with the category *car model*, and a shareholder puts it in the category *stock exchange*. In our opinion the capability to handle the integration and cooperation of information sources taking into account how they are perceived by the user could be obtained by enriching the SDR network with features specifically conceived for handling how a user perceives the information source she/he visits. We argue that

this is an extremely challenging issue which could enable the construction of suitable personalized front-ends to a large set of systems such as data and web warehouses, cooperative information systems, e-commerce sites, web portals, etc., which are capable of adapting themselves to the profiles of the user accessing them.

Acknowledgements. The authors thank Luigi Palopoli for many inspiring discussions about the arguments of the paper.

References

Arens Y, Knoblock CA, Chee CY, et al. (1993) Retrieving and integrating data from multiple information sources. International Journal of Cooperative Information Systems 2(2):127–158

Batini C, Lenzerini M (1984) A methodology for data schema integration in the entity relationship model. IEEE Transactions in Software Engineering 10(6):650–664

Batini C, Castano S, Fugini MG, et al. (1995) Tecniche per l'analisi di descrizioni di processi nella pubblica amministrazione. In *Atti del Congresso Annuale dell'AICA* (*AICA'95*), Cagliari, Italy, pp 247–258 (in Italian)

Bergamaschi S, Castano S, Vincini M (1999) Semantic integration of semistructured and structured data sources. SIGMOD Record 28(1):54–59

Bergamaschi S, Castano S, Vincini M, et al. (2001) Semantic integration and query of heterogeneous information sources. Data & Knowledge Engineering 36(3):215–249

Bernstein PA, Rahm E (2000) Data warehouse scenarios for model management. In Proceedings of the international conference on conceptual modeling (ER'00), Salt Lake City, UT. Lecture Notes in Computer Science, Springer, Berlin, pp 1–15

Buitelaar P, Van De Riet RP (1992) The use of a lexicon to interpret er-diagrams: a like project. In Proceedings of the international conference on the entity-relationship approach, (ER'92), Karlsruhe, Germany. Lecture Notes in Computer Science, Springer, Berlin, pp 162–177

Calvanese D, De Giacomo G, Lenzerini M, et al. (1998) Description logic framework for information integration. In Proceedings of the international conference on principles of knowledge representation and reasoning (KR'98), Trento, Italy. Morgan Kaufmann, San Mateo, CA, pp 2–13

Castano S, De Antonellis V (1997) Semantic dictionary design for database interoperability. In Proceedings of the international conference on data engineering (ICDE '97), Birmingham, UK. IEEE Computer Society, Los Alamitos, CA, pp 43–54

Castano S, De Antonellis V, De Capitani di Vimercati S (2001) Global viewing of heterogeneous data sources. Transactions in Data and Knowledge Engineering 13(2):277–297

Collet C, Huhns MN, Shen WM (1991) Resource integration using a large knowledge base in carnot. IEEE Computer 24(12):55–62

Doan A, Domingos P, Halevy A (2001) Reconciling schemas of disparate data sources: a machine-learning approach. In Proceedings of the international conference on management of data (SIGMOD 2001), Santa Barbara, CA. ACM Press, New York

Ellmer E, Huemer C, Merkl D, et al. (1995) Neural network technology to support view integration. In Proceedings of the international conference on object-oriented and entity-relationship modelling (OOER'95), Gold Coast, Australia. Lecture Notes in Computer Science, Springer, Berlin, pp 181–190

Fankhauser P, Kracker M, Neuhold EJ (1991) Semantic vs. structural resemblance of classes. ACM SIGMOD RECORD 20(4):59–63

Flesca S, Palopoli L, Saccà D, et al. (1998) An architecture for accessing a large number of autonomous, heterogeneous databases. Networking and Information Systems Journal 1(4–5):495–518

Garcia-Molina H, Papakonstantinou Y, Quass D, et al. (1997) The TSIMMIS approach to mediation: data models and languages. Journal of Intelligent Information Systems 8:117–132

Gotthard W, Lockemann PC, Neufeld A (1992) System-guided view integration for object-oriented databases. IEEE Transactions on Knowledge and Data Engineering 4(1):1–22

Haas LM, Miller RJ, Niswonger B, et al. (1999) Transforming heterogeneous data with database middleware: beyond integration. IEEE Data Engineering Bulletin 22(1):31–36

Hayne S, Ram S (1990) Multi-user view integration system (muvis): an expert system for view integration. In Proceedings of the international conference on data engineering (ICDE '90), Los Angeles, CA. IEEE Computer Society, Los Alamitos, CA, pp 402–409

Johannesson P (1993) Using conceptual graph theory to support schema integration. In Proceedings of the international conference on the entity-relationship approach (ER'93), Arlington, TX. Lecture Notes in Computer Science, Springer, Berlin, pp 283–296

Levy A, Rajaraman A, Ordille J (1996) Querying heterogeneous information sources using source descriptions. In Proceedings of the international conference on very large data bases (VLDB'96), Bombay, India. Morgan Kaufmann, San Mateo, CA, pp 251–262

Madhavan J, Bernstein PA, Rahm E (2001) Generic schema matching with cupid. In Proceedings of the international conference on very large data bases (VLDB 2001), Rome, Italy. Morgan Kaufmann, San Mateo, CA, pp 49–58

Metais E, Meunier JN, Levreau G (1993) Database schema design: a perspective from natural language techniques to validation and view integration. In Proceedings of the international conference on conceptual modeling (ER'93), Dallas TX. Lecture Notes in Computer Science, Springer, Berlin, pp 190–205

Metais E, Kedad Z, Comyn-Wattiau I, et al. (1997) Using linguistic knowledge in view integration: toward a third generation of tools. Data & Knowledge Engineering 23(1):59–78

Miller AG (1995) WordNet: a lexical database for English. Communications of the ACM 38(11):39–41

Milo T, Zohar S (1998) Using schema matching to simplify heterogenous data translations. In Proceedings of the international conference on very large data bases (VLDB'98), New York. Morgan Kaufmann, San Mateo, CA, pp 122–133

Mirbel I (1995) Semantic integration of conceptual schemes. In Proceedings of the international workshop on applications of natural language to data bases (NLDB'95), Versailles, France. AFCET

Mitra P, Wiederhold G, Jannink J (1999) Semi-automatic integration of knowledge sources. In Proceedings of Fusion'99, Sunnyvale, CA.

Navathe SB, Elmasri R, Larson JA (1986) Integrating user views in database design. IEEE Computer 19(1):50–62

Palopoli L, Pontieri L, Terracina G, et al. (1999) Semi-automatic construction of a data warehouse from numerous large databases. In Proceedings of the international conference on re-technologies for information systems (ReTIS'00), Zurich, Switzerland. Osterreichische Computer Gesellschaft, pp 55–75

Palopoli L, Pontieri L, Terracina G, et al. (2000) Intensional and extensional integration and abstraction of heterogeneous databases. Data & Knowledge Engineering 35(3):201–237

Palopoli L, Terracina G, Ursino D (2001b) A graph-based approach for extracting terminological properties of elements of XML documents. In Proceedings of the international conference on data engineering (ICDE 2001), Heidelberg, Germany. IEEE Computer Society, Los Alamitos, CA, pp 330–340

Pontieri L, Ursino D, Zumpano E (2002) An approach for synergically carrying out intensional and extensional integration of data sources having different formats. In Proceedings of the international conference on advanced information systems engineering (CAiSE 2002), Toronto, Ontario. Lecture Notes in Computer Science, Springer, Berlin, pp 752–756

Rahm E, Bernstein PA (2001) A survey of approaches to automatic schema matching. VLDB Journal 10(4):334–350

Richardson SD, Dolan WB, Vanderwende L (1998) Mindnet: acquiring and structuring semantic information from text. In Proceedings of the international conference on computational linguistics (COLING-ACL'98), Montreal, Quebec. Morgan Kaufmann, San Mateo, CA, pp 1098–1102

Rosaci D, Terracina G, Ursino D (2001) Deriving 'sub-source' similarities from heterogeneous, semi-structured information sources. In Proceedings of the IFCIS conference on cooperative information systems (CoopIS 2001), Trento, Italy. Lecture Notes in Computer Science, Springer, Berlin, pp 163–178

Roth MT, Schwarz PM (1997) Don't scrap it, wrap it! A wrapper architecture for legacy data sources. In Proceedings of the international conference on very large data bases (VLDB 1997), Athens, Greece. Morgan Kaufmann, San Mateo, CA, pp 266–275

Sheth AP, Larson JA, Cornelio A, et al. (1998) A tool for integrating conceptual schemata and user views. In Proceedings of the international conference on data engineering (ICDE'88), Los Angeles, CA. IEEE Computer Society, Los Alamitos, CA, pp 176–183

Spaccapietra S, Parent C (1994) View integration: a step forward in solving structural conflicts. IEEE Transactions on Knowledge and Data Engineering 6(2):258–274

Terracina G, Ursino D (2000) Deriving synonymies and homonymies of object classes in semi-structured information sources. In Proceedings of the international conference on management of data (COMAD 2000), Pune, India. McGraw-Hill, New York, pp 21–32

Ursino D (2000) Extraction and exploitation of intensional knowledge from heterogeneous information sources. PhD thesis, Lecture Notes in Computer Science 2282, Springer, Heidelberg

Wiederhold G (1992) Mediators in the architecture of future information systems. IEEE Computer 25(3):38–49

Appendix: Constructing an SDR Network from Pre-existing Data Sources

A.1. Constructing an SDR Network from an XML Document

In defining rules for constructing an SDR network $Net(D) = \langle NS(D), AS(D) \rangle$ from an XML document D, it is useful to assume the availability of the corresponding DTD. This produces a speed-up for the construction process; however, it is not strictly needed. Indeed, all DTD information necessary for obtaining the SDR network can be inferred directly from the XML document with some additional computational costs.

A.1.1. Definition of the SDR Network Nodes

Recall that the set $NS(IS)$ of nodes of the SDR network $Net(IS)$, associated with an information source IS, is the union of two sets $NS_C(IS)$ and $NS_A(IS)$, where $NS_C(IS)$ denotes complex nodes, whereas $NS_A(IS)$ indicates atomic nodes.

In order to obtain $NS(D)$, we examine the DTD associated with D. Each node in $NS_C(D)$ is obtained from a non-terminal element of the DTD. Vice versa, each node of $NS_A(D)$ is derived from either an attribute or a terminal element of the DTD, with the following exceptions:

1. *IDREF* and *IDREFS* attributes do not have an associated node since they denote links between elements and, therefore, they are directly represented by arcs in $Net(D)$.
2. If two or more attributes of different elements have the same name, all of them are represented in $Net(D)$ by a unique node in $NS_A(D)$.
3. If an attribute has the same name as an (either terminal or non-terminal) element, it is represented in $Net(D)$ by the node associated with that element.[7]

Finally, in order to derive the number of instances associated with each node, it is necessary to count the instances relative to the corresponding concepts within the XML document.

A.1.2. Definition of the SDR Network Arcs

In this section we illustrate how the set of arcs $AS(D)$ of $Net(D)$ can be derived from D. In the following we use the same symbol to indicate an element (respectively, an attribute) of D and the corresponding node in $Net(D)$.

$AS(D)$ is the union of the sets $AS_N(D)$ and $AS_R(D)$. Arcs belonging to $AS_N(D)$, called *nesting arcs*, are obtained from nesting relationships. An arc $\langle S, T, L_{ST} \rangle$ is added to $AS_N(D)$ if either: (i) T is a sub-element of the element S, or (ii) T is an attribute associated with the element S and its type is different from both *IDREF* and *IDREFS*. Therefore, $AS_N(D)$ can be obtained by examining the DTD only.

Arcs belonging to $AS_R(D)$, called *reference arcs*, are derived from *IDREF* and $IDREFS$ attributes. In particular, given an *IDREF* or an *IDREFS* attribute of an element S, referencing one or more instances of an element T, we associate a *reference arc* from the SDR network node S to the SDR network node T. Since $IDREF$ and $IDREFS$

[7] Recall that, in the DTD, two elements cannot have the same name.

attributes are untyped, only the source nodes of the arcs of $AS_R(D)$ can be derived from the DTD; for determining the target nodes, we clearly need to analyze the 'data component', that is, the XML document.

A.1.3. Definition of the Labels Associated with the SDR Network Arcs

Consider an arc $\langle S, T, L_{ST} \rangle$ belonging to $AS(D)$ (recall that $L_{ST} = [d_{ST}, r_{ST}]$, where d_{ST} is the semantic distance coefficient and r_{ST} is the semantic relevance coefficient). In the following, we use the term 'concept' to indicate elements and attributes defined in a DTD and we call *objects* the corresponding instances represented in the associated XML document.

Definition A.1. Let D be an XML document. An object O' is a *component* of an object O if either (i) O is the instance of an element E, O' is the instance of an element E' and E' is a sub-element of E, or (ii) O is the instance of an element E and O' is the instance of an attribute A of E.

The function $comp(O)$ can then easily be defined, which, given an object O, returns the set of its components.

Definition A.2. Let D be an XML document and let $Net(D)$ be the corresponding SDR network. Let N be a node of $Net(D)$. The *XML-Assoc-ObjSet* of N is the set of instances of the concept which N has been derived from.[8]

Consider now the following sets of nodes:

- NS_S denoting the XML-Assoc-ObjSet of S;
- NS_T denoting the XML-Assoc-ObjSet of T;
- $RNS_{S,T}$ denoting the set of objects $O_i \in NS_S$ such that, for at least one object $O \in NS_T$, $O \in comp(O_i)$.

The semantic distance coefficient d_{ST} for S and T is defined as:

$$d_{ST} = \frac{\sum_{O_i \in RNS_{S,T}} \gamma_{xml}(O_i, T)}{|RNS_{S,T}|}$$

where:

[8] Note that, in some situations, N may be obtained from two or more concepts (e.g., when an attribute and an element have the same name). In this case, the XML-Assoc-ObjSet of N consists of the set of instances of all concepts which N has been derived from.

$$\gamma_{xml}(O_i, T) = \begin{cases} 0 & \text{if } \exists p \text{ such that } p \in NS_T,\ p \in comp(O_i),\ p \text{ is an attribute different from } IDREF \text{ or } IDREFS \text{ and } \nexists q \text{ such that} \\ 0.25 & q \in NS_T, q \in comp(O_i), q \neq p \text{ if } \exists p \text{ such that } p \in NS_T, p \in comp(O_i),\ p \text{ is a terminal element and } \nexists q \text{ such that } q \in NS_T, q \in comp(O_i), q \neq p, q \text{ is either an element or the object referred to by an } IDREF \text{ or } IDREFS \text{ attribute of } O_i \\ 0.5 & \text{if } \exists p, q \text{ such that } p \in NS_T, q \in NS_T,\ p \in comp(O_i), q \in comp(O_i),\ p \neq q,\ p \text{ and } q \text{ are terminal elements and } \nexists r \text{ such that } r \in NS_T, r \in comp(O_i), r \text{ is either a non-terminal element or the object referred to by an } IDREF \text{ or } IDREFS \text{ attribute of } O_i \\ 1 & \text{if } \exists p \text{ such that } p \in NS_T,\ p \in comp(O_i) \text{ and either } p \text{ is a non-terminal element or } p \text{ is the object referred by an } IDREF \text{ or } IDREFS \text{ attribute of } O_i \end{cases}$$

The reasoning underlying the definition of d_{ST} is as follows: an attribute directly defines a property of an element and, therefore, is part of the element itself; as a consequence the semantic distance associated with it is 0. A terminal sub-element directly defines the concept associated with it; vice versa a non-terminal sub-element defines the concept associated with it by means of the set of its sub-elements. Thus, given an object O (belonging to $RNS_{S,T}$), a terminal sub-element O' (belonging to both NS_T and $comp(O)$) is semantically closer to O than a non-terminal sub-element O'' (belonging to both NS_T and $comp(O)$). Moreover, if two or more terminal sub-elements with the same name are components of the same element O, we can conclude that one of them alone is not enough to completely specify a given property of O whereas they, as a whole, do specify this property. In this case the semantic distance between each of these terminal elements and O is intermediate w.r.t. the distances defined for single terminal elements and non-terminal ones. $IDREF$ and $IDREFS$ attributes generally represent relationships between elements; thus the semantic distance between an element O, having an $IDREF$ or an $IDREFS$ attribute, and the referred one can be assumed to be the same as that defined between non-terminal elements.

As far as the *semantic relevance coefficient* is concerned, we have:

$$r_{ST} = \frac{|RNS_{S,T}|}{|NS_S|}$$

This formula directly derives from the definition of the semantic relevance of T w.r.t. S as the participation degree of the concept associated with T in defining the concept associated with S, that is, the fraction of instances of the concept denoted by S whose complete characterization requires at least one instance of the concept represented by T.

In several cases r_{ST} can be immediately obtained by exploiting the following property.

Property A.1. The semantic relevance coefficient r_{ST} is equal to 1 when either T is a *#REQUIRED* attribute, or T is an attribute having a default value, or T is a sub-element which an operator $+$ is associated with in the DTD.

```
<!DOCTYPE University [
<!ELEMENT Professor (Phone+, e-mail*, Project*, OfficeAddress)>
    <!ATTLIST Professor
       ID ID #REQUIRED
       Name CDATA #REQUIRED
       Birthdate CDATA #IMPLIED
       Birthplace CDATA #IMPLIED
       Teaches_in IDREFS #IMPLIED
       Papers IDREFS #IMPLIED
    >
    <!ELEMENT Phone (#PCDATA)>
    <!ELEMENT e-mail (#PCDATA)>
    <!ELEMENT Project (#PCDATA)>
    <!ATTLIST Project
       Starting_Date CDATA #IMPLIED
       Ending_Date CDATA #IMPLIED
    >
    <!ELEMENT OfficeAddress(#PCDATA)>
    <!ATTLIST OfficeAddress
       Road CDATA #REQUIRED
       City CDATA #REQUIRED
    >
    <!ELEMENT Paper (#PCDATA)>
    <!ATTLIST Paper
       ID ID #REQUIRED
       Authors IDREFS #REQUIRED
       Type (Journal|Conference) ''Journal''
       Volume CDATA #REQUIRED
       Pages CDATA #REQUIRED
    >
    <!ELEMENT Course (Year?, Student_Number?, Argument+)>
<!ATTLIST Course
    ID ID #REQUIRED
    Name CDATA #REQUIRED
    Responsible IDREF #REQUIRED
    Propaedeutic_Courses IDREFS #IMPLIED
    Propaedeutic_For IDREFS #IMPLIED
>
<!ELEMENT Year (#PCDATA)>
<!ELEMENT Student_Number (#PCDATA)>
<!ELEMENT Argument (#PCDATA)>
<!ELEMENT Student (Address, Thesis?)>
<!ATTLIST Student
    ID ID #REQUIRED
    Name CDATA #REQUIRED
    Birthdate CDATA #IMPLIED
    Birthplace CDATA #IMPLIED
    Enrollment_Year CDATA #IMPLIED
    Exams IDREFS #IMPLIED
    Attended_Courses IDREFS #IMPLIED
>
<!ELEMENT Address (#PCDATA)>
<!ELEMENT Thesis (Title, Topic+)>
<!ELEMENT Title (#PCDATA)>
<!ELEMENT Topic (#PCDATA)>
<!ELEMENT Exam (Date, Grade)>
<!ATTLIST Exam
    ID ID #REQUIRED
    Student IDREF #REQUIRED
    Course IDREF #REQUIRED
>
<!ELEMENT Date (#PCDATA)>
<!ELEMENT Grade (#PCDATA)>
]>
```

Fig. 10. An XML document representing the web site of a university.

The time complexity for constructing an SDR network from an XML document is quadratic in the number of instances. Vice versa, the space complexity for carrying out the same task is constant.

In order to construct an SDR network from an XML document no human intervention is required. Indeed, all the rules for carrying out such a task are based on the structure of both the DTD and the corresponding XML document; as a consequence, these rules may be executed by an automatic parser.

As an example, consider the XML document DTD U_X depicted in Fig. 10 and representing a web site relative to a university (due to space limitations we do not show the corresponding XML document). The corresponding SDR network U_{X_SDR} is illustrated in Fig. 1.

A.2. Constructing an SDR Network from an E/R Scheme

The entity-relationship model is quite different from both the OEM and the XML conceptual models. Indeed, it has been designed for representing *structured information sources*; in addition, while in XML and OEM conceptual models the relationships are implicitly represented as labeled references, the E/R model represents them *explicitly*; moreover, *they can have attributes*. Finally, the E/R model provides *is-a* relationships which do not directly correspond to any construct of either XML or OEM.

In the translation rules that we are presenting below, we try to achieve the simplest possible structure of the obtained SDR network. In particular, whenever possible, we translate relationships simply as arcs connecting nodes corresponding to involved entities (this cannot be done for many-to-many relationships with attributes).

Analogously to the previous cases, the process of constructing the SDR network $Net(ER) = \langle NS(ER), AS(ER)\rangle$ corresponding to an E/R scheme ER consists of three steps that derive nodes, arcs and labels of the SDR network, respectively. In the following, without loss of generality, we can assume that all n-ary relationships occurring in an

input E/R scheme have been suitably transformed into binary ones. In addition, when referring to the SDR network resulting from the translation of an E/R scheme, we will use the term 'entity' (respectively, 'relationship', 'attribute') also for denoting the SDR network node corresponding to an entity (respectively, a relationship, an attribute).

A.2.1. Definition of the SDR Network Nodes

Note that there exists a semantic correspondence holding for E/R attributes, OEM atomic nodes and XML attributes as well as among E/R entities, OEM complex nodes and XML non-terminal elements. Therefore, the rules for obtaining the set $NS(ER)$ of nodes of the SDR network $Net(ER)$, corresponding to an E/R scheme ER, are analogous to those used for deriving the sets $NS(D)$ and $NS(G)$ of nodes of the SDR networks $Net(D)$ and $Net(G)$, corresponding to an XML document D illustrated in Section A.1.1. In particular, the set $NS(ER)$ consists of the union of two sets of nodes:

$$NS(ER) = NS_C(ER) \cup NS_A(ER)$$

where $NS_C(ER)$ is the set of complex nodes whereas $NS_A(ER)$ is the set of atomic nodes. A node of $NS_A(ER)$, named M, is created for each set of attributes in ER sharing the same name, say M. A node of $NS_C(ER)$ is created for each entity or for each many-to-many relationship with attributes occurring in ER.

In order to obtain the number of instances associated with each node, it is necessary to consider the instances relative to the corresponding concepts within the associated databases.

As a consequence of the reasoning of Section A.2, in order to achieve the simplest possible structure for the resulting SDR network, both one-to-one and one-to-many and many-to-many relationships without attributes occurring in ER are not represented by nodes in $Net(ER)$.

A.2.2. Definition of the SDR Network Arcs and Labels

Let ER be an E/R scheme and let $Net(ER)$ be the corresponding SDR network. The rules for obtaining the set $AS(ER)$ of SDR network arcs of $Net(ER)$ and the corresponding labels have been defined by taking into account that (i) an E/R scheme represents a structured information source and, consequently, all entity instances have the same structure (and this simplifies the computation of labels of $AS(ER)$); (ii) both one-to-one and one-to-many and many-to-many relationships without attributes are represented by SDR network arcs.

In more detail, the arcs of $Net(ER)$ and the corresponding labels are constructed by applying the following rules:

- There is an arc from an entity to each of its attributes; this arc has the semantic distance coefficient set equal to 0 and the semantic relevance coefficient set equal to 1 since each instance of an entity is associated with exactly one instance of the attribute.
- If R is a relationship from E_i to E_j without attributes, it is represented by both an arc A_{ij} from E_i to E_j and an arc A_{ji} from E_j to E_i. The semantic distance coefficient of both A_{ij} and A_{ji} is 1 whereas the semantic relevance coefficient of A_{ij} (respectively, A_{ji}) is equal to the fraction of instances of E_i (respectively, E_j) connected to at least one instance of E_j (respectively, E_i).
- If R is either a one-to-one or a one-to-many relationship with attributes, it can be translated as in the previous case.

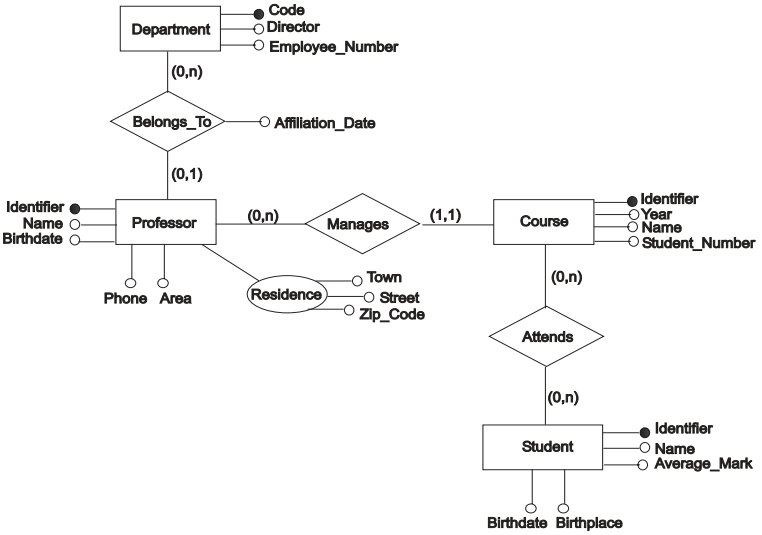

Fig. 11. The E/R diagram $U_{E/R}$ relative to a university.

- If R is a many-to-many relationship with attributes, it is represented by an SDR network complex node. In $Net(ER)$, there is an arc A_{ir} from E_i to R, an arc A_{rj} from R to E_j, an arc A_{jr} from E_j to R and an arc A_{ri} from R to E_i. The semantic distance coefficients of all these arcs are 1; the semantic relevance coefficient of A_{rj} (respectively, A_{ri}) is 1 since each instance of R is linked to exactly one instance of E_j (respectively, E_i). The semantic relevance coefficient of A_{jr} (respectively, A_{ir}) is equal to the fraction of instances of E_j (respectively, E_i) taking part in R.

Finally, let E_i and E_j be two entities of ER such that an *is-a* relationship exists from E_j to E_i. This particular relationship is translated in $Net(ER)$ by a pair of arcs: the first one, from E_j to E_i, has a semantic distance coefficient equal to 0, since the knowledge of E_i is necessary for characterizing E_j (e.g., attributes of E_i are inherited by E_j), and a semantic relevance coefficient equal to 1, since each instance of E_j requires, for its complete definition, an instance of E_i. The second arc, from E_i to E_j, has a semantic distance coefficient equal to 1, since the knowledge of E_j is not necessary for characterizing E_i, and a semantic relevance coefficient equal to the ratio between the number of instances of E_j and that of E_i; indeed, this quantity represents the fraction of instances of E_i whose complete definition requires at least one instance of E_j.

The time complexity for constructing an SDR network from an XML document is linear in the number of instances. The space complexity for performing the same activity is constant.

The construction of an SDR network from an E/R scheme depends only on the structural characteristics of the scheme; as a consequence, it can be automatically carried out by a parser and no human intervention is required.

As an example, consider the E/R scheme $U_{E/R}$, depicted in Fig. 11 and representing the conceptual scheme of a university database. The corresponding SDR network U_{E_SDR} is illustrated in Fig. 3.

Author Biographies

Domenico Rosaci received the Laurea Degree in Civil Engineering from the University of Reggio Calabria in February 1994. From September 1994 to January 1996 he was a member of the Computer Science group at DIMET. From January 1996 to January 1999 he was a PhD student at University of Reggio Calabria. He received a PhD in Electronic Engineering in March 1999. From October 1999 he has been a Research Assistant at the Mediterranean University of Reggio Calabria. His research interests include knowledge extraction and representation, information source integration and abstraction, cooperative information systems, data warehouses, data compression and histograms, and intelligent agents.

Giorgio Terracina received the Laurea Degree in Computer Engineering from the University of Calabria in April 1999. From May 1999 to March 2000 he was a member of the Knowledge Engineering group at DEIS. From March 2000 he has been a PhD student at the Mediterranean University of Reggio Calabria. His research interests include knowledge extraction and representation, scheme integration and abstraction, cooperative information systems, data warehouses and semi-structured data.

Domenico Ursino received the Laurea Degree in Computer Engineering from the University of Calabria in July 1995. From September 1995 to January 1997 he was a member of the Knowledge Engineering group at DEIS. From January 1997 to January 2000 he was a PhD student at University of Calabria. From January 2000 to October 2000 he was a member of the Knowledge Engineering group at DEIS and of the database group at ISI-CNR. From October 2000 he has been an Assistant Professor at the Mediterranean University of Reggio Calabria. His research interests include knowledge extraction and representation, information source integration and abstraction, cooperative information systems, data warehouses, and intelligent agents.

Correspondence and offprint requests to: Domenico Ursino, DIMET – Università Mediterranea di Reggio Calabria, Via Graziella, Loc. Feo di Vito, 89060 Reggio Calabria, Italy. Email: ursino@ing.unirc.it