Towards a Comprehensive Methodological Framework for Semantic Integration of Heterogeneous Data Sources*

D. Calvanese[♯], S. Castano[†], F. Guerra[§], D. Lembo[♯], M. Melchiori[‡], G. Terracina[¶], D. Ursino[¶], M. Vincini[§]

[♯] DIS – Università di Roma "La Sapienza" – e-mail {calvanese,lembo}@dis.uniroma1.it

[†] DSI – Università di Milano – e-mail castano@dsi.unimi.it

[‡] DEA – Università di Brescia – e-mail melchior@ing.unibs.it

[§] DSI – Università di Modena e Reggio Emilia – e-mail {guerra,vincini}@dsi.unimo.it

[¶] DIMET – Università di Reggio Calabria – e-mail {terracina,ursino}@ing.unirc.it

Abstract

Nowadays, data can be represented and stored by using different formats ranging from non structured data, typical of file systems, to semistructured data, typical of Web sources, to highly structured data, typical of relational database systems. Therefore, the necessity arises to define new models and approaches for uniformly handling datasources having different formats and structures, and obtaining a global, integrated, and uniform representation. In this paper we present three approaches to data integration and propose a unifying framework integrating the various methodologies and incorporating techniques developed separately. We also present the architecture of a metadata repository supporting the integration framework.

1 Introduction

Recent developments of information and communication technologies enable accessing a large number of structured and semistructured datasources, developed at different times, with different organizational principles and models, and supported by different hw/sw platforms. Moreover, in the last few years, a rich variety of models and languages for representing and manipulating datasources over the Web has been proposed. Indeed, nowadays, data can be represented and stored by using different formats, ranging from non structured data, typical of file systems, to highly structured data, typical of relational database systems, to semistructured data, typical of Web datasources [1, 17, 25].

In this context, comprehensive methodological frameworks and tools for semantic integration are required, in particular to *(i)* handle the enormous quantity of data typical of the Web; *(ii)* deal with highly heterogeneous datasources, in particular with respect to the structuring level of data; datasources must be managed that

*Work partially supported by a MURST grant under the project "D2I".

do not have a precise structure and the cooperation and the uniform treatment of both structured and semi-structured datasources must be guaranteed; *(iii)* provide support for expressing queries in terms of global views over underlying datasources, and conceive mechanisms for the reformulation and/or answering of such queries in terms of the data stored in the sources.

The latter problem is known in the literature as view-based query processing, and has been studied very actively in the recent years [27, 18, 12].

In this paper, we address the above problems and we describe the initial results carried on within an ongoing Italian national research project, called *D2I* (Integration, Warehousing, and Mining of Heterogeneous Data Sources). The goal of this project is the definition of a comprehensive methodological framework for the integration, warehousing, and mining of heterogeneous sources and the development of specific results for the following three tasks: *(i)* semantic integration of data coming from heterogeneous sources, *(i)* data warehouse design and querying, and *(i)* data mining. In this paper we concentrate on the first task, and we describe the initial architecture of the integration framework for uniformly and semi-automatically handling datasources having great dimensions and different formats and structures. The framework puts into a coherent whole a set of approaches and techniques developed as the first step of the *D2I* project by the four different partners involved in the integration tasks of *D2I*, namely the Univ. of Calabria, the Univ. of Milano, the Univ. of Modena and Reggio Emilia, and the Univ. of Rome. The first approach is based on graphs and extends to semistructured data, the techniques developed in the DIKE system [2, 22, 21]. The second approach is object-oriented and extends to semistructured data the techniques developed in the MOMIS system [7, 4, 13]. Finally, the third approach is based on a two level architecture, in which a logical representation of sources in terms of the relational data model is mapped to a conceptual representation of the domain of interest, given in terms of the expressive Description Logic $\mathcal{DLR}$ [9, 10]. These approaches are compatible and their common features can be summarized as follows:

- Semantically rich representation of involved datasources through a common conceptual model which is exploited for deriving and representing the semantics of each involved datasource.
- Semiautomatic extraction of interschema properties relating concepts (or subschemas) of datasource at the conceptual level. Since the number and the dimension of datasources are great and since the information they store change quite frequently over time, manual extraction of interschema properties is expensive and difficult.
- Construction of an integrated and unified representation of the involved datasources, to be used for querying the integration system, and explicit representation of the mapping between the data at the sources and the integrated representation;
- Use of Description Logics due to their formalization and reasoning capabilities in both semantic integration and view-based query processing.

In this paper, we first present the three approaches, which include a variety of techniques to support semantic integration as well as query processing over integrated representations. For each of the three approaches, we have defined a metadata schema describing the metadata maintained during the integration activities.

Second, we propose a comprehensive framework for integration based on a common *metadata repository* architecture, which constitutes a unification layer for the various approaches. On the one hand, this allows us to migrate to the common framework the various techniques developed independently in the three approaches, in order to obtain a unique, well structured, and detailed integration methodology. On the other hand, the common framework and its metadata repository will constitute the reference basis for data warehousing and data mining applications and techniques developed in the project.

The paper is organized as follows. In Section 2, 3 and 4 we describe the three different approaches to representation and integration of heterogeneous datasources proposed by the Univ. of Calabria (in collaboration with the Univ. of Reggio Calabria), the Univ. of Milano in collaboration with the Univ. of Modena and Reggio Emilia, and the Univ. of Rome respectively. Section 5 presents the common metadata repository architecture, and Section 6 concludes the paper.

2 The DIKE Approach

In this section we describe the DIKE approach jointly developed by the University of Calabria and the University of Reggio Calabria [21]. This approach provides semi-automatic techniques for the extraction and the representation of interschema properties as well as the integration of datasources having different formats and structures.

In order to uniformly handle and represent heterogeneous datasources, DIKE exploits a conceptual model called SDR-Network [22, 26]. Given a datasource DS, the associated SDR-Network $Net(DS)$ is a rooted labeled graph $Net(DS) = \langle NS(DS), AS(DS)\rangle$. Here, $NS(DS)$ is a set of nodes, each one representing a concept of DS. Each node is identified by a name indicating the concept it represents. $AS(DS)$ denotes a set of arcs; an arc represents a relationship between two concepts. More specifically, an arc 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. L_{ST} is a pair $[d_{ST}, r_{ST}]$, where both d_{ST} and r_{ST} are coefficients belonging to the real interval $[0, 1]$. d_{ST} is the *semantic distance coefficient*; it indicates 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 to T to characterize the concept associated to S. r_{ST} is the *semantic relevance coefficient*, representing the fraction of instances of the concept denoted by S whose complete definition requires at least one instance of the concept denoted by T. Semantic preserving translations have been provided from some interesting source formats, such as XML, OEM and ER to SDR-Network [22, 26].

2.1 Interschema Property Extraction

The DIKE approach for deriving interschema properties consists of a technique for deriving synonymies and homonymies between concepts and a technique for extracting sub-source similarities.

The technique for extracting synonymies and homonymies among concepts belonging to two heterogeneous datasources DS_1 and DS_2 first determines the similarity degree of each pair of concepts $C_l \in DS_1$ and $C_m \in DS_2$ and then derives synonymies and homonymies. The similarity degree of a pair of concepts $C_l \in DS_1$ and $C_m \in DS_2$ depends on the similarity of the neighborhoods of C_l and C_m; these are determined according to a suitable metrics based on the semantic distance and semantic relevance coefficients of the SDR-Network; the closer to C_l and C_m the neighborhoods are, the stronger their influence is. The similarity of a pair of neighborhoods is computed by constructing a suitable bipartite graph from the concepts of the neighborhoods into consideration and by computing a maximum weight matching on it. The set of significant synonymies (resp., homonymies) is then constructed by selecting those pairs of concepts whose similarity degree is greater (resp., smaller) than a certain, dynamically computed threshold th_{Syn} (resp., th_{Hom}). All details about DIKE technique for extracting synonymies and homonymies can be found in [22].

The second technique aims at deriving sub-source similarities. Given a datasource DS and the corresponding SDR-Network $Net(DS)$, the number of possible sub-sources that can be identified in DS is exponential in the number of nodes of $Net(DS)$. To avoid the burden of analyzing such a huge number of sub-sources, our technique first selects only the most *promising* ones according to empirical rules. After this, it determines the similarity degree associated to each pair of promising sub-sources in a way analogous to that used for deriving concept similarities. All details about the technique for extracting sub-source similarities can be found in [24].

2.2 Datasource Integration

The algorithm for datasource integration receives two SDR-Networks DS_1 and DS_2 and integrates them for obtaining a global SDR-Network SDR_G. The algorithm first juxtaposes DS_1 and DS_2 for constructing a (temporarily redundant and, possibly, ambiguous) global SDR-Network SDR_G. In order to normalize SDR_G, by removing its inconsistencies and ambiguities, several transformations must be carried out on it. SDR_G normalization is composed by the following sub-steps:

- *SDR-Network node examination.* Each pair of synonym nodes $N_x \in DS_1$ and $N_y \in DS_2$ are assumed to coincide in SDR_G and, therefore, must be merged into a new node N_{xy}. Each pair of homonym nodes $N_p \in DS_1$ and $N_q \in DS_2$ 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 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-source examination.* Each pair of similar sub-sources $SS_x \in DS_1$ and $SS_y \in DS_2$ must be "merged". The merge of sub-sources could lead to the presence

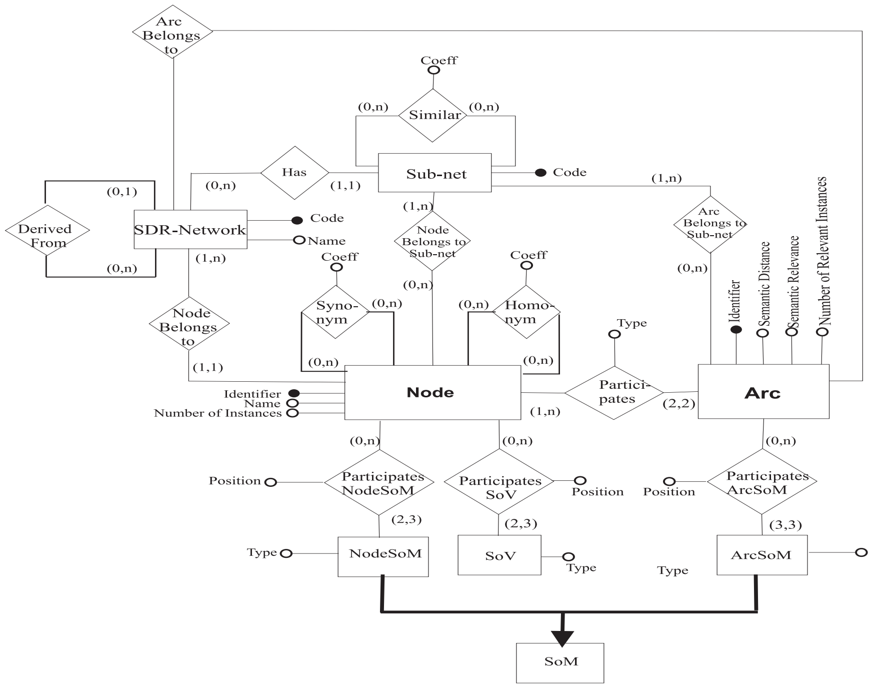

Figure 1: The DIKE Metadata Repository Architecture

of pairs of arcs connecting the same pair of nodes; if this happens, the arcs composing the pair must be merged.

The set of transformations the algorithm carries out are stored in a log called *Set of Mappings*; this describes the way a node (resp., an arc) of SDR_G has been obtained from one or more nodes (resp., arcs) belonging to input SDR-Networks. From the Set of Mappings the algorithm derives a *Set of Views*; this allows to obtain instances of nodes of SDR_G from instances of nodes of DS_1 and DS_2. All details about the integration algorithm outlined above can be found in [23].

The ER schema of the architecture of metadata repository supporting the DIKE approach is shown in Figure 1. It stores information about involved SDR-Networks, interschema properties and the set of transformations carried out during the integration process. In particular, the entities *Node*, *Arc*, *SDR-Network* and *Sub-net* are exploited for storing the involved datasources. Significant synonymies (resp., homonymies, subsource similarities) derived during the interschema property extraction step are represented by the cyclic relationship *Synonym* (resp., *Homonym*, *Similar*). Finally, the entities *SoM* and *SoV* are used for storing the *Set of Mappings* and the *Set of Views*.

3 The MOMIS Approach

In this section, we describe the MOMIS integration approach jointly developed by the University of Milano and the University of Modena and Reggio Emilia [3, 7]. This approach provides semiautomatic techniques for extraction and representation of interschema properties and for schema clustering and integration, to identify candidates to integration and synthesize candidates into an integrated global schema.

The MOMIS integration process is based on a source independent object-oriented model called ODM_{I^3} used for describing structured and semistructured data sources in a common way. ODM_{I^3} derives from ODMG-ODM model with the following extensions: i) the union constructor, to express alternative data structures in the definition of an ODM_{I^3} class, capturing requirements of semistructured data; ii) the optional constructor to specify that an attribute is optional for an instance; iii) integrity constraint rules in order to express, in a declarative way, *if then* integrity constraint rules at both intra and intersource level; iv) mapping rules in order to express relationships between the global schema description and the schema description of the original sources. From ODM_{I^3} model we have derived the corresponding ODL_{I^3} language. For semistructured datasources, schema description is generally not directly available and is specified directly within data [8]. In this case, object patterns are first extracted from the source and are then translated in ODL_{I^3}. ODL_{I^3} description of XML datasources is defined by considering document type descriptions associated with the source (e.g., DTDs). A ODL_{I^3} compatible formalism for the representation of DTDs for integration is described in [14].

3.1 Interschema Property Extraction and Representation

The first phase of the integration process has the goal extracting and representing interschema properties. A *Common Thesaurus* of terminological intensional and extensional relationships is constructed, describing interschema knowledge about ODL_{I^3} classes and attributes of source schemas. In the Common Thesaurus, interschema knowledge is expressed through intensional and extensional relationships. Intensional relationships are SYN (Synonym-of), BT (Broader Terms) and its inverse NT (Narrower Terms), and RT (Related Terms). They are defined between classes and attributes, and are specified by considering class/attribute names. Intensional relationships SYN, BT and NT between two classes may be "strengthened" by establishing that they are also *extensional* relationships. Consequently, SYN_{ext}, BT_{ext}, and NT_{ext} extensional relationships can be defined in ODL_{I^3}.

The Common Thesaurus is built through an incremental process during which relationships are added in the following order: *i) schema-derived relationships*: intensional relationships holding at intraschema level are extracted by analyzing each ODL_{I^3} schema separately; *ii) lexical-derived relationships*: intensional relationships holding at interschema level are extracted by analyzing different sources ODL_{I^3} schemas together according to the Wordnet supplied ontology; *iii) designer-supplied relationships*: intensional and extensional relationships are supplied directly by the designer, to capture domain knowledge about the source schemas. Supplied relationships are validated with ODB-Tools [5]; *iv) inferred relationships*: a new set of terminological relationships is inferred by ODB-Tools by reasoning over the union of the local schemas enriched with available relationships and by deriving new generalization and aggregation properties.

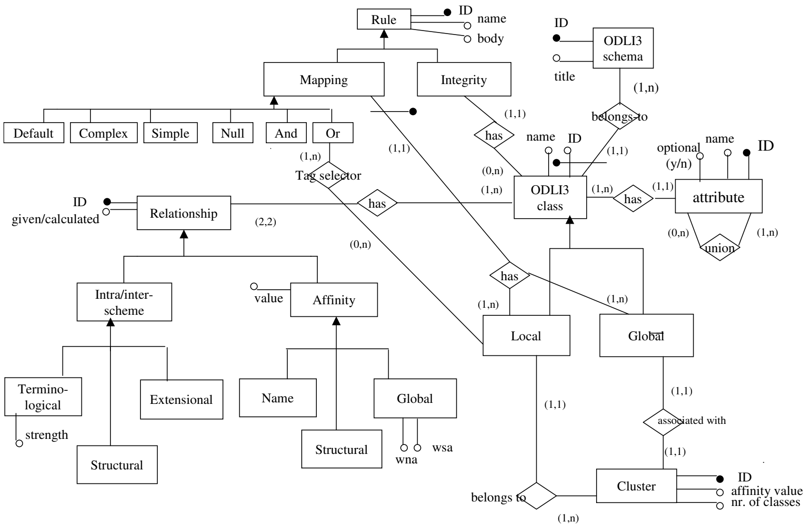

Figure 2: The MOMIS Metadata Repository Architecture

3.2 Schema Clustering and Integration

Goal of this phase of the integration process is to identify ODL_{I^3} classes that describe the same or semantically related information in different source schemas and to integrate them into global ODL_{I^3} classes. To integrate ODL_{I^3} classes of the different sources into global ODL_{I^3} classes, we employ hierarchical clustering techniques based on the concept of *affinity*. This activity is performed with the ARTEMIS tool environment [13]. *Affinity coefficients* (i.e., numerical values in the range $[0, 1]$) are evaluated for all possible pairs of ODL_{I^3} classes, based on the (valid) terminological relationships in the Common Thesaurus properly strengthened. Affinity coefficients determine the degree of semantic relationship of two classes based on their names (Name Affinity coefficient) and their attributes (Structural Affinity coefficient). A comprehensive value of affinity, called *Global Affinity* coefficient, is finally determined as the linear combination of the Name and Structural Affinity coefficients. Global affinity coefficients are then used by a hierarchical clustering algorithm, to classify ODL_{I^3} classes according to their degree of affinity. The output of the clustering procedure is an affinity tree, where ODL_{I^3} classes are the leaves and intermediate nodes have an associated affinity value, holding for the classes in the corresponding cluster. Clusters for integration are interactively selected from the affinity tree using a threshold based mechanism. For each selected cluster in the tree, a global class providing the unified view of all the classes of the cluster is defined. The generation of global classes is interactive with the designer. Selected a cluster in the affinity tree, first, a set of global attributes, corresponding to the union of the attributes of the classes belonging to the cluster, is defined. Unification of local attribute names is performed automatically by using terminological relationships holding for them in the Common Thesaurus (e.g., for attributes that have a SYN relationship, only one term is selected as the name for the corresponding global attribute in the global class). To complete global class definition,

information on attribute mappings and default values is provided by the designer in the form of *mapping rules*. For each global class a persistent *mapping-table* storing all the intensional mappings is generated. It is a table whose rows represent the set of the local classes which belong to the corresponding cluster and whose columns represent the global attributes. An element of this table represents how the global attribute ag is mapped to a local class L.
The ER schema of the metadata repository architecture supporting the MOMIS integration approach is reported in Figure 2.

4 The $\mathcal{DLR}$ Approach

In this section, we describe the $\mathcal{DLR}$ integration approach, developed at the University of Rome. In $\mathcal{DLR}$, a data integration system is modeled at two different levels:

- The *conceptual level* contains a conceptual representation of the data managed by the system, including a conceptual representation of the data residing in each source, a conceptual representation of the global concepts and relationships that are of interest to the enterprise and have been currently analyzed, and an explicit declarative account of the conceptual interdependencies among the data in different sources. The conceptual level constitutes the global schema that provides a consolidated view of the information to the outside of the system.

- The *logical level* contains a representation in terms of a logical data model of the sources and of the answers to queries posed to the integration system (which may be possibly materialized and maintained by the system). Specifically, we adopt the relational model, and assume that each source is represented by a set of relations. Non-relational data sources are presented by suitable wrappers in the relational format.

Below we first describe more in detail the conceptual level, and then we relate the two levels and describe query processing in $\mathcal{DLR}$.

4.1 The Conceptual Level Specification

The conceptual level is expressed in terms of the expressive description logic $\mathcal{DLR}$ [10, 9], which allows for representing the domain of interest by means of *concepts*, which denote classes of objects, n*-ary relationships*, which denote sets of tuples, each of which represents an association between conceptual objects, and *attributes*, which associate properties to conceptual objects (or tuples of conceptual objects). Each attribute value belongs to one of several *domains*, and intensional relationships between domains can be specified by means of *domain assertions*, stating inclusion between the corresponding sets of values.

Complex concept and relationship expressions can be built applying suitable *constructs* starting from atomic concepts and relationships. Such constructs allow for expressing boolean operators on concepts and relationships, specifying the type of relationship components, and imposing cardinality constraints on the participation to relationships. For the precise syntax and semantics of the $\mathcal{DLR}$ constructs we refer to [9]. An important aspect of the conceptual representation is the explicit specification of the set of interdependencies between the elements of the conceptual model, either source or enterprise elements. In this respect, the specification of such interdependencies expressed over the conceptual level can be regarded as the process of understanding and representing the relationships between data residing in different datasources and the information of interest to the enterprise. The $\mathcal{DLR}$ approach does not propose a specific methodology to extract such interdependencies from source representations. However, due to the basic similarities between the conceptual model adopted in the other approaches presented in this paper and $\mathcal{DLR}$, the techniques for semi-automatic property extraction, proposed in the other approaches, can be applied also to $\mathcal{DLR}$ [20].

Formally, the global schema is constituted by a set of *assertions*, which have the form $C_1 \sqsubseteq C_2$ or $R_1 \sqsubseteq R_2$, where C_1 and C_2 are $\mathcal{DLR}$ concepts, and R_1 and R_2 are $\mathcal{DLR}$ relations of the same arity.

The semantics of a schema is given by specifying when a database satisfies the constraints in the schema. Formally, a database $\mathcal{DB}$ is a set of relations, one relation $L^{\mathcal{DB}}$ for each concept or relationship L in the schema (such relations are obtained from the relations associated to atomic concepts and relationships, according to the semantics of the $\mathcal{DLR}$ constructs). A database $\mathcal{DB}$ *satisfies* an assertion $L_1 \sqsubseteq L_2$ if $L_1^{\mathcal{DB}} \subseteq L_2^{\mathcal{DB}}$, and it satisfies a schema if it satisfies all assertions in the schema.

$\mathcal{DLR}$ allows one to capture virtually every conceptual data model, including the ER model, UML, and Object-oriented data models, also augmented with several forms of constraints that usually cannot be expressed in such models.

4.2 Data Integration and Query Processing

Integrating heterogeneous datasources consists in providing a uniform access to the sources in terms of a common representation. In the $\mathcal{DLR}$ approach such a representation is given by the conceptual level, in terms of which the queries to the integration system are formulated. Such queries must be answered using the data residing at the sources, and to do so it is necessary to specify how the source relations at the logical level relate to the elements of the conceptual level.

In the $\mathcal{DLR}$ approach such a mapping is specified by associating to each source-relation a view over the conceptual level, thus following the local-as-view approach [27, 18]. Such a view is expressed as a non-recursive Datalog query in which the predicates in the atoms are concepts, relationships, attributes, and domains of the conceptual level. Notice that such predicates can be arbitrary $\mathcal{DLR}$ relationships and concepts, freely used in the assertions at the conceptual level. This distinguishes the $\mathcal{DLR}$ approach with respect to [15, 16, 19], where no constraints can be expressed at the conceptual level on the concepts and relationships that appear in the queries.

To actually compute the answer to a query over the conceptual level, the approach followed in $\mathcal{DLR}$ is to reformulate such a query in terms of the source relations, i.e., solve the query rewriting problem [27, 18]. However, due to the heterogeneity of the sources one must also take into account that the same data in different sources may

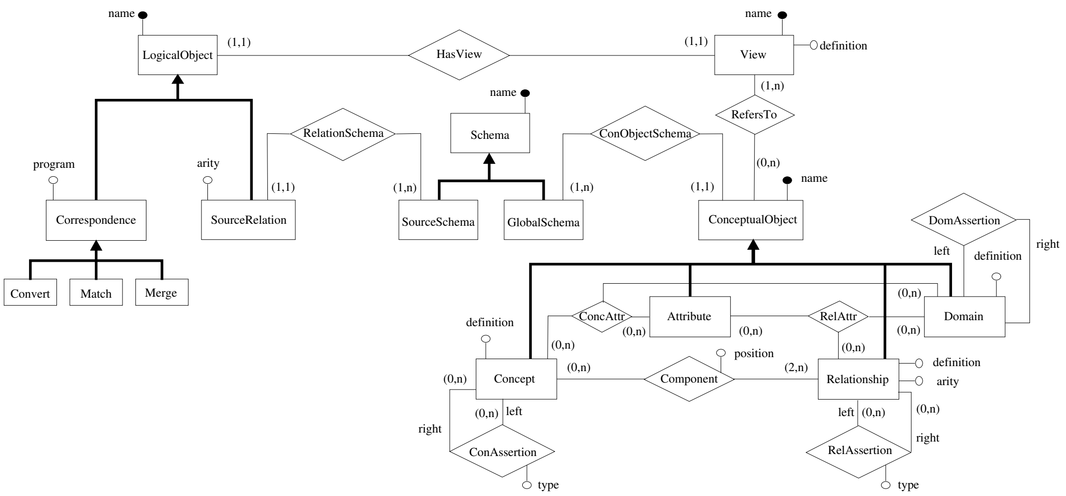

Figure 3: The $\mathcal{DLR}$ Metadata Repository Architecture

be represented in different ways. To deal with this aspect, the $\mathcal{DLR}$ approach introduces so called *Reconciliation Correspondences*, which specify how data in different sources can *match*, how data in different sources can be *merged* or data in a single source can be *converted* to produce the answers to queries [11]. Each Reconciliation Correspondence is expressed as a (non materialized) view over the conceptual level, which specifies the condition under which the Correspondence is applicable, and has an associated program, which performs the appropriate (matching, merging, or conversion) operation on the actual data. Such Correspondences are properly taken into account by the rewriting algorithm [11].

In Figure 3 we report a simplified ER model of the metadata repository architecture which supports the $\mathcal{DLR}$ approach to data integration.

5 The Common Metadata Repository

The three approaches to data integration described in the above sections, even though individually developed by the different partners involved in the *D2I* project, present several similar features:

- the use of a global schema expressed in a conceptual data model, which provides a unified and reconciled view of the information at the sources and which can be queried by the user.
- the use of a mapping between the source schema and the global schema. The MOMIS and the DIKE systems adopt the *global-as-view* (GAV) approach, in which each global element is defined in terms of the elements of the source schemas, which are represented in a conceptual data model. The $\mathcal{DLR}$ system adopts the *local-as-view* (LAV) approach, in which the source schemas, expressed in the relational model, are defined in terms of the global elements [27];

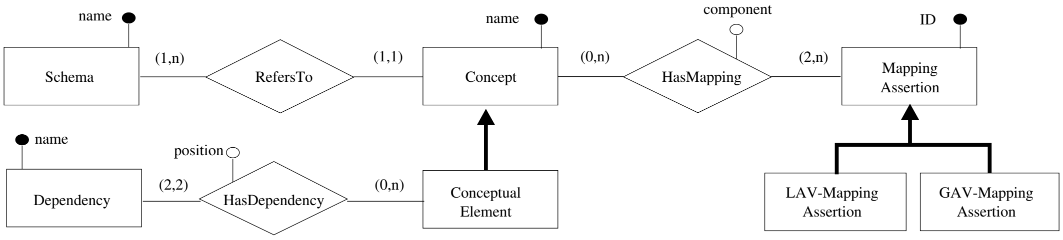

Figure 4: The General Metadata Repository Architecture

- the specification and/or semiautomatic extraction of interschema properties among concepts/sub-schemas of the source schema (MOMIS and DIKE), or among the source elements at the conceptual level ($\mathcal{DLR}$).

Based on such similarities we propose a common framework which supports the activities of the three approaches. In figure 4 we present the ER schema of the general metadata repository architecture, designed to store the representation of integration applications in terms of the common framework mentioned above. The entity *Concept* represents a generic piece of information, either stored at the sources or maintained in the global schema. The relationship *RefersTo* relates each global *Concept* or source *Concept* with the *Schema* it belongs to. The entity *Mapping Assertion* represents either the LAV or the GAV assertions that establish the relationships between the information stored at the sources and the information represented in the global schema. The relationship *HasMapping* indicates the information involved in each mapping assertion. The cardinality constraint $(2,n)$ on the participation of *Mapping Assertion* in *HasMapping* states that each mapping assertion involves at least two instances of *Concept*. The attribute *component* assigns a position to each *Concept* appearing in the mapping statement. The entity *Dependency* represents properties/assertions at the conceptual level, expressed between elements at the conceptual level, either in the global schema or in the source schemas. The attribute *position* on the relationship *HasDependency* is used to distinguish the two elements involved in a dependency.

The metadata repository architecture described above represents the common characteristics of the three repositories individually developed to support the activities of the $\mathcal{DLR}$, the DIKE, and the MOMIS approaches. The main components of each single repository are specializations of the components of the general one. For example, the entities *Node* and *Arc* of the DIKE repository, $ODLI_3$ *Class* of the MOMIS repository, and *ConceptualObject*, *View*, and *SourceRelation* of the $\mathcal{DLR}$ repository are specializations of the entity *Concept*. Hence, we conceive the common repository architecture as the union of the general repository and of the $\mathcal{DLR}$, the DIKE, and the MOMIS metadata repositories, together with the proper ISA relationships between related entities and relationships (see [6] for details).

The metadata repository provides a centralized and unified representation of the metadata documenting all activities related to integration applications. Thus, it will be used as a common basis by the various tools developed within the D2I project aiming at supporting the integration tasks.

6 Conclusions

In this paper we have described three approaches to the representation, extraction, and integration of heterogeneous datasources, developed by the different partners involved in the project *D2I*. The common aspects are the use of a conceptual model common to all sources, the definition of interschema properties relating data in different sources, and the use of Description Logics to formalize the conceptual component of an integration system and reason about it. The three approaches have concentrated on different aspects and problems related to the construction of integration systems, but the basic similarities between them allow for the definition of a comprehensive framework for semantic integration and the migration to it of techniques developed separately. We have presented the definition of the architecture of a common metadata repository, which is the first step towards an integration of the various features of the three approaches.

Currently, under the project *D2I*, we are working on further unifying the three approaches and developing a common methodology incorporating the techniques for the various integration steps developed separately by the different partners. We still have to detail the functional interface of the metadata repository, also taking into account the necessity to access the repository from a Description Logic reasoner. This allows us to automatically deduce, and subsequently store in the repository, relevant properties related to the integration activity, such as consistency or redundancy of (portions of) conceptual schemas.

A further aspect requiring investigation concerns the datamodel in which to implement the metadata repository. Both a structured model (e.g., relational) or a semistructured model (e.g., XML-schema) could be chosen, although a semistructured implementation seems to be more appropriate. Indeed, while the portion of the metadata repository supporting integration is well structured, we are also extending the metadata architecture towards supporting warehousing and mining activities, and the related metadata present a semistructured form.

References

[1] S. Abiteboul. Querying semi-structured data. In *Proc. of International Conference on Database Theory (ICDT'97)*, Lecture Notes in Computer Science, pages 1–18. Springer-Verlag, 1997.

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

[3] D. Beneventano, S. Bergamaschi, S. Castano, A. Corni, R. Guidetti, G. Malvezzi, M. Melchiori, and M. Vincini. Information integration: the MOMIS project demonstration. In *International Conference on Very Large Data Bases (VLDB 2000)*, 2000.

[4] D. Beneventano, S. Bergamaschi, F. Guerra, and M. Vincini. The MOMIS approach to information integration. In *AAAI International Conference on Enterprise Information Systems (ICEIS 2001)*, 2001.

[5] D. Beneventano, S. Bergamaschi, C. Sartori, and M. Vincini. ODB-QOPTIMIZER: A tool for semantic query optimization in oodb. In *Int. Conference on Data Engineering (ICDE'97)*, 1997.

[6] D. Beneventano et al. Specification for the metadata repository. Technical Report D0.R1, D2I Project – Integration, Warehousing, and Mining of Heterogeneous Data Sources, 2001.

[7] S. Bergamaschi, S. Castano, D. Beneventano, and M. Vincini. Retrieving and integrating data from multiple sources: the MOMIS approach. *Data and Knowledge Engineering*, 36:251–249, 2001.

[8] P. Buneman. Semistructured Data. In *Proc. of Symposium on Principles of Database Systems (PODS'97)*, pages 117–121, 1997.

[9] D. Calvanese, G. De Giacomo, and M. Lenzerini. On the decidability of query containment under constraints. In *Proc. of the 17th ACM SIGACT SIGMOD SIGART Symp. on Principles of Database Systems (PODS'98)*, pages 149–158, 1998.

[10] D. Calvanese, G. De Giacomo, M. Lenzerini, D. Nardi, and R. Rosati. Description logic framework for information integration. In *Proc. of the 6th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR'98)*, pages 2–13, 1998.

[11] D. Calvanese, G. De Giacomo, M. Lenzerini, D. Nardi, and R. Rosati. Data integration in data warehousing. *Int. J. of Cooperative Information Systems*, 2001. To appear.

[12] D. Calvanese, G. De Giacomo, M. Lenzerini, and M. Y. Vardi. What is query rewriting? In *Proc. of the 7th Int. Workshop on Knowledge Representation meets Databases (KRDB 2000)*, pages 17–27, 2000.

[13] S. Castano, V. D. Antonellis, and S. D. C. di Vimercati. Semantic integration of heterogeneous data sources. *IEEE Transactions on Data and Knowledge Engineering*, 13(2), 2001.

[14] S. Castano, V. D. Antonellis, S. D. C. di Vimercati, and M. Melchiori. An XML-based framework for information integration over the Web. In *Proc. of Int. Workshop on Information Integration and Web-based Applications & Services 2000*, Yogyakarta, Indonesia, 2000.

[15] F. M. Donini, M. Lenzerini, D. Nardi, and A. Schaerf. A hybrid system integrating Datalog and concept languages. In *Proc. of the 2nd Conf. of the Ital. Assoc. for Artificial Intelligence (AI*IA'91)*, volume 549 of *Lecture Notes in Artificial Intelligence*. Springer-Verlag, 1991.

[16] F. M. Donini, M. Lenzerini, D. Nardi, and A. Schaerf. $\mathcal{AL}$-log: Integrating Datalog and description logics. *J. of Intelligent Information Systems*, 10(3):227–252, 1998.

[17] R. Goldman, J. McHugh, and J. Widom. From semistructured data to XML: Migrating the lore data model and query languages. In *Proc. of International Workshop on the Web and Databases (WebDB'99)*, pages 25–30, 1999.

[18] A. Y. Halevy. Theory of answering queries using views. *SIGMOD Record*, 29(4):40–47, 2000.

[19] A. Y. Levy and M.-C. Rousset. CARIN: A representation language combining Horn rules and description logics. In *Proc. of the 12th Eur. Conf. on Artificial Intelligence (ECAI'96)*, pages 323–327, 1996.

[20] L. Palopoli, D. Saccà, and D. Ursino. Semi-automatic techniques for deriving interscheme properties from database schemes. *Data and Knowledge Engineering*, 30(3):239–273, 1999.

[21] L. Palopoli, G. Terracina, and D. Ursino. The system DIKE: Towards the semi-automatic synthesis of cooperative information systems and data warehouses. In *Proc. of Symposium on Advances in Databases and Information Systems (ADBIS-DASFAA 2000)*, pages 108–117, 2000.

[22] L. Palopoli, G. Terracina, and D. Ursino. A graph-based approach for extracting terminological properties of elements of XML documents. In *Proc. of International Conference on Data Engineering (ICDE 2001)*, pages 330–340. IEEE Computer Society, 2001.

[23] D. Rosaci, G. Terracina, and D. Ursino. An algorithm for obtaining a global representation from information sources having different nature and structure. In *Proc. of International Conference on Database and Expert Systems Applications (DEXA 2001)*, 2001.

[24] D. Rosaci, G. Terracina, and D. Ursino. Deriving "sub-source" similarities for information sources having different structure and nature. Submitted for publication. Available from the authors., 2001.

[25] D. Suciu. Semistructured data and XML. In *Proc. of International Conference on Foundations of Data Organization (FODO'98)*, 1998.

[26] G. Terracina and D. Ursino. Deriving synonymies and homonymies of object classes in semi-structured information sources. In *Proc. of International Conference on Management of Data (COMAD 2000)*, pages 21–32. McGraw Hill, 2000.

[27] J. D. Ullman. Information integration using logical views. In *Proc. of the 6th Int. Conf. on Database Theory (ICDT'97)*, volume 1186 of *Lecture Notes in Computer Science*, pages 19–40. Springer-Verlag, 1997.