Journal of Universal Computer Science, vol. 9, no. 5 (2003), 423-446
submitted: 6/5/02, accepted: 18/3/03, appeared: 28/5/03 © *J.UCS*

Cyclical Structure Converter(CSC): a System for Handling the Interaction of Structured and Semi-structured Data Sources

Jameson Mbale
(Department of Computer Science, Harbin Institute of Technology
92 Xidhazi Street, Nangang District, Box 773, Harbin 150001, China
mbalej@www.com)

Domenico Ursino
(DIMET, Università "Mediterranea" di Reggio Calabria
Via Graziella, Località Feo di Vito, 89060 Reggio Calabria, Italy
ursino@unirc.it)

Xu Xiao Fei
(Department of Computer Science, Harbin Institute of Technology
92 Xidhazi Street, Nangang District, Box 773, Harbin 150001, China
xiaofei@hope.hit.edu.cn)

Abstract: This paper aims at investigating the integration of structured data into a semi-structured environment. In particular, it introduces the Cyclic Structure Converter (CSC) system that performs this task. In CSC, correspondence assertions and integration rules provide the adequate intelligence to reconcile the (possible) heterogeneous semantics relative to involved information sources. CSC has also the capability to filter and process only the relevant operational data. CSC's versatility in maneuvering with different data models allows it to be applied into any field, such as engineering, insurance, medicine, space science and education, to mention a few.

Key Words: Information Source Integration, Interschema Property, Cooperative Information Systems, Semi-structured Information Sources
Category: H.2.4, H.3.4

1 Introduction

1.1 Motivations

The technology that has made DBMS's possible is the direct result of successful programs in Computer Science research, performed in the past couple of decades. One of the most relevant researches in DBMS technology was to make possible the construction of a global schema, storing information of different databases belonging to a certain application field. The integration activity facilitates a global access to a group of heterogeneous resources, hence allowing interoperability among various organizations.

Although DBMS evolution has been capable of successfully facing a large variety of exigencies, it continuously needs to evolve in order to fit the new challenges. Among these, interaction with the World Wide Web appears to be one of the most interesting. Data on the Web are handled by semi-structured representation formats whose features are quite different from those characterizing the structured paradigms, typical of databases. In order to allow cooperation between databases and the Web, the need arises for new tools and methods capable of handling databases based on the semi-structured paradigm.

The management of semi-structured information sources has attracted many researchers and various approaches have been proposed for successfully handling such a task. Our approach has been conceived to operate in such an application context and aims at equipping the translators and the integrators with enough intelligence to be able to dynamically handle data from various models. The manipulated data will be automatically linked to the semi-structured environment.

1.2 General characteristics of the approach

This paper presents a new system, called *Cyclical Structure Converter* (CSC), capable of uniformly managing both structured and semi-structured data sources.

The CSC architecture is shown in Figure 1. It allows basic information sources to be constructed and handled by different data representation formats. The data source schemas are extracted and passed to a component named TIM (*Translator, Integrator and Matcher*). This receives correspondence assertions and integration rules from human experts and exploits them to manage data stored in the involved information sources. In particular, TIM returns the various mappings existing among objects belonging to different sources.

These mappings are then passed to the RMA (*Relevant Metadata Attribute*) Extractor that extracts relevant metadata useful to daily operations of the organization. The human expert is again required for setting out the parameters determining the data to be extracted. The criteria used to guide the extraction are based on the frequent daily usage data, whereby the human expert is able to figure out the query composition.

The extracted metadata are sent to the *RMA Graph Builder* which constructs a global representation of all involved data sources; this is represented and managed by a specific graph-based conceptual model, called *SDR-Network* [Terracina and Ursino 2000].

The Global SDR-Network is, then, forwarded to the *Query Formulator* component. This receives some parameters such as object patterns, sub-query object patterns and references from the human expert (see below).

The formulated query is delivered to the *Query Decomposer*, where the global query is decomposed by exploiting references associated with object patterns.

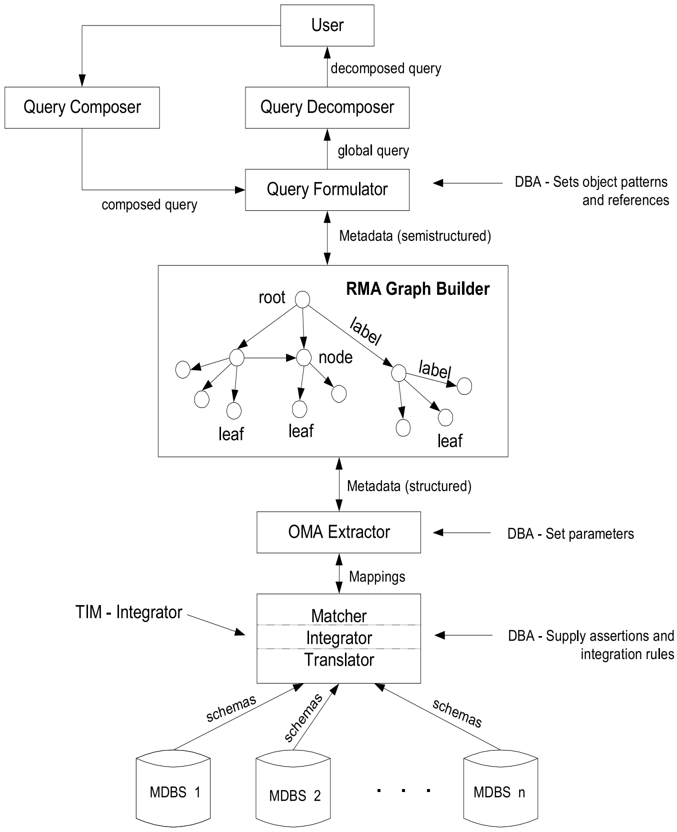

Figure 1: The Cyclical Structure Converter architecture

Queries obtained from the decomposition are, then, executed and obtained answers are collected by a module named *Query Composer*. The global answer is, then, sent to the *Query Formulator*. As a consequence, there is a cycle among the *Query Formulator*, the *Query Decomposer*, the *User* and the *Query Composer*. Hence, the system assumes its name as Cyclical Structure Converter (CSC).

1.3 Related Work

In the past decades, the desire to allow interoperability among different information sources led to the development of the database integration theory; this concentrated mainly on structured information sources.

Internet development made the usage of Web-based semi-structured information sources very popular.

[Abiteboul 1997] points out that semi-structured data models have been intensively studied in recent years; however, in the past, the emphasis has been on topics related to the design of schemas for semi-structured data [Buneman et al. 1997] and in the extraction of schemas from the available data [Goldman and Widom 1997, Nestorov et al. 1998]. In fact, most of the researchers are now motivated with the desire to seek a solution that allows computer users to access the heterogeneous sources on the Web in an integrated form. In these circumstances, a lot of semi-structured data source management tools have been proposed such as $W3QS$ [Konopnicki and Shumueli 1995, Konopnicki and Shumueli 1997] and WedSQL [Mendelzon et al. 1996]. Some of these tools have an origin in TSIMMIS and BDFS that are briefly discussed below.

TSIMMIS (the Stanford-IBM Manager of Multiple Information Sources) [Garcia-Molina 1997] is a project about a self describing model, called Object Exchange Model (OEM) [Abiteboul et al. 1997], that deals with data objects and pattern matching techniques for performing a pre-defined set of queries based on a query template. [Garcia-Molina 1997] emphasizes that the TSIMMIS architecture was initially developed to design tools for facilitating the integration of structured and semi-structured heterogeneous data. Hence TSIMMIS is to be considered as one of the first attempts at developing methodologies for semi-structured data source integration. The Object Exchange Model exploited in TSIMMIS is a self-describing data model having data items associated with descriptive labels without any strong typing system; the semantic knowledge is effectively encoded in the Mediator Specification Language (MSL) rules by enforcing source integration at the mediator level. Even if the generality and conciseness of OEM and MSL makes the TSIMMIS approach a suitable methodology for the integration of widely heterogeneous and semi-structured information sources, there is a major setback in the approach because the dynamic addition of sources is an expensive task. Another disadvantage of TSIMMIS is that it can execute only pre-defined queries and each source modification must be perfomed by manually rewriting the mediator rules.

In [Buneman et al. 1997] the Basic Data Model for Semi-structured Data (BDFS) is introduced. This is an elegant graph-based data model that exploits graphs for representing both portions of a database (called ground graphs) and schemas. In the former case edges are labeled by data; in the latter case edges are labeled by formulae of a suitable logic theory.

[Ouksel and Naiman 1994] illustrates the system SCOPE aiming at reconciling the semantics of heterogeneous sources. SCOPE exploits thesauruses and ontologies for identifying interschema properties, i.e., structural and semantic relationships linking concepts belonging to different schemas; these are, then, represented as assertions. The dynamic and query-oriented integration is, then, performed by manipulating the corresponding assertions. Therefore, the whole process is actually based on the knowledge acquired during the reconciliation process. CSC and SCOPE are similar in that both of them exploit interschema correspondence assertions for harmonizing information sources having different formats.

MOMIS is a system for the integration and querying of information sources. It follows a "semantic approach" to information integration based on an intensional study of information sources and on the following architectural elements: *(i)* a common object-oriented data model used for representing involved information sources; *(ii)* a set of wrappers for translating schema description in the common data model; *(iii)* a mediator and a query processor based on two pre-existing tools, namely ARTEMIS [Castano et al. 2001] and ODB-Tools [Beneventano et al. 1997]. In order to integrate involved information sources, MOMIS constructs a Common Thesaurus that plays the role of a shared ontology for them. The built structure is exploited to determine the degree of affinity associated with pairs of objects belonging to different information sources. The schema integration is then realized by means of a cluster procedure that exploits derived affinity degrees to determine groups of similar objects. The result of the integration procedure is a global flat schema representing all involved information sources.

[Arens et al. 1993] proposes SIMS that exploits Description Logics for creating a global schema definition. Garlic [Roth and Schwarz 1997] exploits a complex wrapper architecture and the language GDL for handling a set of local sources in such a way as to unify them and to produce a global schema. Note that both SIMS and Garlic use a global schema to support all possible user queries on involved schemas; in other words, they have not been conceived for handling only a pre-defined set of queries.

[Goldman and Widom 1997] describes DataGuides; these are a concise, accurate and convenient summary of a set of semi-structured data sources. They are concise because they describe each label path of a source exactly once, regardless of the number of times it appears in the source; they are accurate because they do not encode any label path that does not appear in the sources. They are convenient because a DataGuide is an OEM object that can be manipulated by applying the OEM techniques. The most important drawback of the DataGuides consists in its Query Formulation mechanism, which is difficult if the database structure is not known. Moreover, whenever a source database change, the DataGuides must be updated and this is a cumbersome process.

In the literature a large variety of approaches has been proposed for carrying out schema matching activities in order to derive terminological relationships and to exploit them for guaranteeing information source integration and interoperability ([Parent and Spaccapietra 1995, Scheuermann et al. 1998] and, more recently, [Rahm and Bernstein 2001]).

[Larson et al. 1989] proposes a method that can be made *automatic*; nevertheless it is based only on attribute knowledge (so it is *based on "syntax" and not on "semantics"*). Our approach aims at being *semantic* but also *automatic*.

[Castano et al. 2002] proposes an approach for the integration of XML documents with the support of interschema properties. In particular, an XML document is translated into a set of elements, called *x-classes*; this representation allows the derivation of synonymies, homonymies and type conflicts existing among concepts belonging to different sources. The knowledge of these properties is exploited for carrying out the integration task; this returns a global set of x-classes that is, in its turn, translated into a global XML document with the support of the user who can choose the structure of the final document.

The approach we are proposing in this paper has some similarities with that described in [Castano et al. 2002]. Indeed, both of them: *(i)* are *rule-based* [Rahm and Bernstein 2001]; *(ii)* derive interschema properties that are, then, exploited for carrying out the integration task.

However, the two approaches have several differences; in particular, [Castano et al. 2002] has been conceived for handling only XML documents whereas our approach is capable of managing information sources with different representation formats.

The approach proposed in [McBrien and Poulovassilis 2001] integrates information sources with different representation formats (e.g., E/R, UML, XML). It behaves as follows: first, involved information sources are translated in a particular, auxiliary, graph-based formalism called HDM; then the translated sources are integrated; the global source thus obtained is, finally, translated into one of the original formats. Both the approach of [McBrien and Poulovassilis 2001] and our own have been conceived for allowing the integration of data sources characterized by a large variety of formats.

In [Milo and Zohar 1998] an approach for translating data from a source format to a target one is described. This approach is quite different from our own in its purposes and perspectives. However, it is interesting in that it performs a semantic schema matching operation appearing quite analogous to our interschema property extraction strategy. In particular, both the approaches exploit the neighborhood affinities for determining the semantic similarity of two objects.

In Cupid [Madhavan et al. 2001], a system for deriving interschema properties among heterogeneous information sources is presented. The approach is the first that considers the interschema property derivation, named "schema matching" by the authors, as a task having an existence on its own, independent of the integration activity. The interschema property derivation is performed by carrying out two kinds of examinations, named linguistic and structure matchings. The authors claim that derived properties can be exploited for integration purposes but do not provide a specific integration technique. Both Cupid and our approach have been conceived for handling a large variety of data source formats. However, there are some differences between them: in particular, Cupid only derives interschema properties whereas our approach first detects interschema properties and, then, exploits them for carrying out the integration task. In addition, since the activities Cupid performs for extracting properties are numerous and sophisticated, the obtained results are more precise than those returned by our approach but the required time and user intervention for carrying out the extraction activity are greater than those needed by our methodology.

[Lim and Ng 2001] describes an approach performing the integration of data sources with different formats. Involved sources are first translated into a graph formalism named HDG. After this, all obtained HDG graphs are integrated; such a task is carried out by determining semantic and structural relationships among objects belonging to different sources. The global representation thus obtained is, finally, translated from HDG to XML. The approach of [Lim and Ng 2001] and our own are similar in that: *(i)* both of them are semantic; *(ii)* in both of them the integration is light, even if the approach of [Lim and Ng 2001] requires a translation phase before the integration activity; *(iii)* both of them exploit a lexical dictionary, in particular WordNet; *(iv)* both of them are almost automatic.

In [Doan et al. 2003] an approach, named LSD (Learning Source Description), for carrying out scheme matching activities, is proposed. Differently from most of the other approaches proposed in the literature, as well as from ours, LSD exploits machine learning techniques for deriving properties. As Cupid [Madhavan et al. 2001], also LSD aims only at extracting interschema properties and does not consider the exploitation of such properties for integration purposes. Interestingly enough, LSD requires quite a heavy user support during the initial phase, for carrying out training tasks. After this phase, no human intervention is required. LSD and our approach differ especially in their purposes; indeed, LSD aims at deriving interschema properties, whereas our approach has been conceived mainly for handling integration activities. In addition, as far as the interschema property derivation is concerned, it is worth observing that LSD is "learner-based", whereas our approach is "rule-based" [Rahm and Bernstein 2001]. Finally, LSD requires a heavy human intervention at the beginning and, then, is automatic; vice versa, our approach requires a minor human intervention during the pre-processing phase but needs a further intervention at the end for validating obtained results.

1.4 Paper outline

The plan of the paper is as follows. Section 2 is devoted to describing the SDR-Network conceptual model; the description of the CSC architecture is the argument of Section 3. Section 4 illustrates the role of correspondence assertions in the CSC system; the technique CSC exploits for carrying out the information source integration is the argument of Section 5. Section 6 is devoted to presenting an example case; in Section 7 the CSC capabilities are analyzed; finally, in Section 8, we draw our conclusions.

2 The SDR-Network conceptual model

The SDR-Network [Palopoli et al. 2001, Terracina and Ursino 2000] is a conceptual model for describing data sources that allows uniform modeling of most existing data representation formats as well as derivation and representation of 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 in 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 *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 schema, 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 that 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 an average of these coefficients. r_{ST} is called *semantic relevance coefficient* and represents the fraction of *instances* of the concept

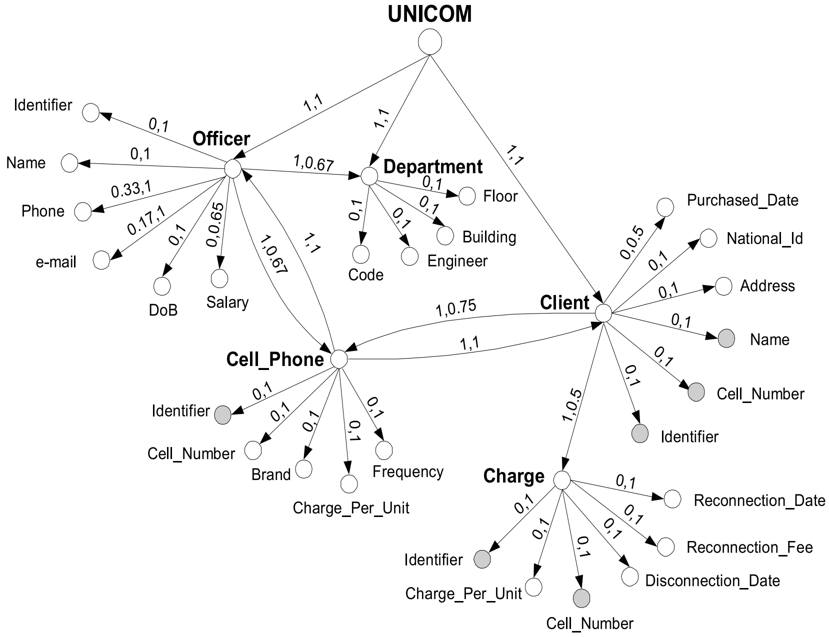

Figure 2: The SDR-Network Net_{MPC} representing a mobile phone company

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 Figure 2. It is relative to a mobile phone company called UNICOM. In the figure, in order to simplify the layout, a grey node named 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 $Cell\_Phone$, $Client$, $Charge$, etc., represent the corresponding concepts. The arc $\langle Officer, Department, [1, 0.67]\rangle$ denotes the existence of a relationship between $Officer$ and $Department$; in particular, it indicates that 67% of officers belong to a department. The other arcs have an analogous semantics.

SDR-Network semantic distance and relevance coefficients can be extended to nodes not directly connected by an arc. This allows us to introduce the notion of node neighborhood that plays a relevant role in the integration of a set of SDR-Network. The neighborhood of a node can be defined as follows.

Definition 1. The *Path Semantic Distance* of a path P in $Net(DS)$ (denoted by PSD_P) is the sum of the semantic distance coefficients associated with the arcs constituting the path.

Definition 2. The *Path Semantic Relevance* of a path P in $Net(DS)$ (denoted by PSR_P) is the product of the semantic relevance coefficients associated with the arcs constituting the path.

Definition 3. A D_Path_n is a path P in $Net(DS)$ such that $n \leq PSD_P < n+1$.

Definition 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 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 1. Consider the node *Client* of the SDR-Network illustrated in Figure 2 (we call this network Net_{MPC} in the rest of the paper). The neighborhoods associated with this node are the following:

$$nbh(Client,0) = \{\langle Client, Identifier, [0,1] \rangle, \langle Client, Cell\_Number, [0,1] \rangle, \\ \langle Client, Name, [0,1] \rangle, \langle Client, Address, [0,1] \rangle, \langle Client, National\_Id, [0,1] \rangle, \\ \langle Client, Purchase\_Date, [0,0.5] \rangle\}$$

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

$$nbh(Client,1) = \{\langle Client, Charge, [1,0.5] \rangle, \langle Charge, Identifier, [0,1] \rangle, \\ \langle Charge, Charge\_Per\_Unit, [0,1] \rangle, \langle Charge, Cell\_Number, [0,1] \rangle, \\ \langle Charge, Disconnection\_Date, [0,1] \rangle, \langle Charge, Reconnection\_Fee, [0,1] \rangle, \\ \langle Charge, Reconnection\_Date, [0,1] \rangle, \langle Client, Cell\_Phone, [1,1] \rangle, \\ \langle Cell\_Phone, Identifier, [0,1] \rangle, \langle Cell\_Phone, Cell\_Number, [0,1] \rangle, \\ \langle Cell\_Phone, Brand, [0,1] \rangle, \langle Cell\_Phone, Charge\_Per\_Unit, [0,1] \rangle, \\ \langle Cell\_Phone, Frequency, [0,1] \rangle$$

For instance, $A = \langle Charge, Reconnection\_Fee, [0,1]\rangle$ belongs to $nbh(Client, 1)$ because $\lfloor Client, Reconnection\_Fee\rfloor_A$ is a D_Path_1 and $Client \neq Reconnection\_Fee$. In a similar fashion, it is possible to derive all the other neighborhoods relative to Net_{MPC}.

Note that, basically, any information source can be represented as a set of concepts and a set of relationships among them. Since SDR-Network nodes and arcs are well suited to represent such concepts and relationships, SDR-Network can be used to uniformly model most existing information sources. In this respect, semantic preserving translation have been provided from some interesting source formats, such as XML, OEM and E/R, to SDR-Networks [Palopoli et al. 2001, Terracina and Ursino 2000].

Presenting all details of the SDR-Network model goes beyond the scope of this paper. The interested reader can find them in [Palopoli et al. 2001, Terracina and Ursino 2000].

3 The CSC architecture

CSC has been developed to achieve the following objectives:

- The enrichment of the local schema semantic representation.
- The identification of the relevant data handled by different DBMS.
- The derivation and the resolution of the schematic and semantic similarities existing among objects.
- The establishment of metadata mappings.
- The capability of handling both structured and semi-structured information.
- The formulation of global queries and their decomposition into sub-queries.
- The application of decomposed sub-queries to meet the demands of the user.
- The query composition to complete the cycle.

CSC, like other systems such as MOMIS, follows the semantic approach in the integration of heterogeneous data built under different models. CSC establishes a method allowing the traditional databases to be automatically converted and integrated in the semi-structured environment. Actually, in the present version, the intervention of the human expert is required to supply some parameters that facilitate the operations performed by some of the system modules.

CSC receives: *(i)* the schemas of the involved information sources; *(ii)* the correspondence assertions and the integration rules involving objects stored therein;

(iii) the parameters necessary for extracting metadata; *(iv)* the object patterns and the references necessary for query formulation.

CSC returns: *(i)* the metadata relative to the involved sources; *(ii)* the mappings among the objects of the global schema and those ones stored in the local schemas; *(iii)* the global queries; *(iv)* the sub-queries derived from the global query decomposition.

The CSC architecture is represented in Figure 1. In the following sub-sections we describe each component into detail.

3.1 Local information sources

These are the basic sources which CSC operates on. They may be characterized by similar or different data models. CSC employs the framework of the SEMINT specific parser [Li and Clifton 2000], which automatically extracts metadata from involved databases. CSC tends to improve this parser in such a way to handle also XML documents and OEM graphs. As in SEMINT, CSC parser is defined as a tool that automatically extracts schema information and constraints from the database catalogues as well as statistics on the data contents using queries over data. The extracted information is classified in three folds, namely: *(i)* attribute names, that compose the dictionary level; *(ii)* schema information, which forms the filed specification level; *(iii)* data contents and statistics, which form the data content level.

3.2 TIM (Translator, Integrator and Matcher)

The TIM module is composed of three sub-modules, namely:

- *The Translator*, which receives the schemas of the local information sources (which could be databases, XML documents and OEM graphs) and constructs the SDR-Network corresponding to them. This is obtained by applying rules described in [Palopoli et al. 2001, Terracina and Ursino 2000]. The human expert is required to define and supply the correspondence assertions and integration rules equipping the whole Translator with the intelligence to resolve and harmonize the information from different models.

- *The Integrator*, which receives the translated schemas, the assertions and the integration rules from the Translator and exploits them for generating a list of correspondences existing among attributes of different data sources.

- *The Matcher*, which receives the attribute correspondences produced by the Integrator and matches the semantically equivalent data elements for producing a mapping in a pair wise form which is, then, delivered to the RMA.

3.3 RMA (Relevant Mapping Attributes) Extractor

The RMA Extractor receives the mappings from the TIM component and extracts only the relevant mappings (metadata) that are useful to the user application. The user is required to supply some parameters allowing to determine the metadata topology to extract. The extracted information is delivered to the RMA Graph Builder.

3.4 RMA Graph Builder

The RMA Graph Builder receives the extracted information from the RMA Extractor and constructs a global SDR-Network representing all the involved information sources. The construction of a global SDR-Network from a set of input SDR-Networks is carried out by applying the methodology described in Section 5. The global SDR-Network thus constructed is then passed to the Query Formulator module.

3.5 Query Formulator

The Query Formulator receives: *(i)* the global SDR-Network from the RMA Graph Builder, *(ii)* a set of object patterns and a set of references from the human expert.

Object patterns represent the structure of data in a semi-structured source; more specifically, an object pattern is defined for each different concept in the source, by considering the set of objects describing it. From a formal point of view, an object pattern is defined as follows:

Definition 6. Let S be a semi-structured source represented by an OEM graph and let $G_k = \{so_1, so_2, \ldots, so_p\}$ be a set of semi-structured objects denoting different instances of a given concept in the source, characterized by the same label l_{so} in S. An object pattern $op_k = \{l_k, A_k\}$ is a pair of the form $op_k = (l_k, A_k)$ where $l_k = l_{so}$ and A_k is the set of attribute labels defined for the objects so_i belonging to G_k.

Example 2. Consider the semi-structured information source

$$IS_1 = MOBILE\_PHONE(brand, name, size, price)$$

The OEM representation of IS_1 is represented in Figure 3.

The object patterns relative to IS_1 are:

- make-pattern = (make, {brand, model})
- brand-pattern = (brand, {name, size, price })

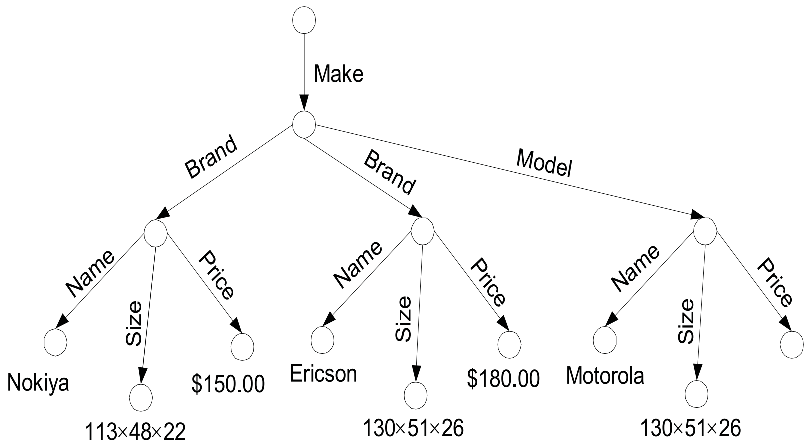

Figure 3: The OEM representation of IS_1

– model-pattern = (model, {name, size, price })

References are used in the object patterns to establish a strong semantic correspondence that will be used during the global query decomposition.

The Query Formulator supports the user to formulate the global queries and sends them to the Query Decomposer.

3.6 Query Decomposer

The Query Decomposer receives the global queries, the object patterns and the sets of references from the Query Formulator and decomposes each global query into a set of sub-queries by exploiting references associated with it. Decomposed queries are, then, passed to the Query Composer.

3.7 Query Composer

The Query Composer receives decomposed queries relative to a given global query, executes them and composes returned results in such a way to obtain a global answer. This is, then, sent to the Query Formulator.

4 Correspondence Assertions in the CSC system

Correspondence assertions are defined as declarative statements asserting that something in one schema is somehow related to something in another schema.

Correspondence assertions could be further illustrated as the hypothetical predefined algebraic sets that best conform to the expected relationships existing among objects in different schemas. Hence, correspondence assertions provide the various components of CSC with enough intelligence to harmonize the (possibly heterogeneous) semantics of involved schemas.

The equipped components are, therefore, extremely versatile; in particular, they can handle the constraints inherited from various data models. In addition, they are dynamically alerted to capture any semantic similarity existing between the input schemas.

In the following we provide some definitions of correspondence assertions.

Definition 7. The *Real World State* of an object type O (resp., a complex or a single attribute A) is the set of real world objects represented by the set of the present occurrences of O (resp., A).

A function RWS is defined that receives an object type O (resp., an attribute A) and returns its Real World State.

Definition 8. Let X_1 (resp., X_2) be an object type (resp., a complex or a single attribute A) belonging to the schema S_1 (resp., S_2). We say that there is a *correspondence assertion* between X_1 and X_2 if:

1. X_1 *and* X_2 *are equivalent*, expressed as $X_1 \equiv X_2$; this happens if, at any time, $RWS(X_1) = RWS(X_2)$;
2. X_1 *contains* X_2, expressed as $X_1 \supseteq X_2$; this happens if, at any time, $RWS(X_1) \supseteq RWS(X_2)$;
3. X_1 *and* X_2 *intersect*, expressed as $X_1 \cap X_2$; this states that, at some time, $RWS(X_1) \cap RWS(X_2) \neq \emptyset$;
4. X_1 *and* X_2 *are disjoint*, expressed as $X_1 \neq X_2$; this states that, at any time, $RWS(X_1) \cap RWS(X_2) = \emptyset$.

Example 3. Consider the following databases:

DB1: MOBILE_PHONE(brand_name, size, weight, call_time, color, price, m_phone)

DB2: CELL_PHONE(br_name, model, battery, m_size, frequency, cost, m_telno, res_phone)

By applying Definitions 7 and 8, it is possible that MOBILE_PHONE $\equiv$ CELL_PHONE, with the following attribute correspondences: *(i)* brand_name = br_name; *(ii)* size = m_size; *(iii)* price = cost; *(iv)* m_phone = m_telno; *(v)* m_phone = res_phone.

Definition 9. Let $X_1\langle cor\rangle X_2$ be a correspondence assertions. Let A_{11}, A_{12}, ..., A_{1n} be the attributes of X_1; let A_{21}, A_{22}, ..., A_{2n} be the attributes of X_2. Let o be any element common to both X_1 and X_2 real world states (i.e., $o \in (RWS(X_1) \cap RWS(X_2)))$; let e_1 and e_2 be the occurrences representing o in the databases described by S_1 and S_2. Then, $X_1\langle cor\rangle X_2$, with corresponding attributes $attcor_1(A_{11}, A_{21})$, $attcor_2(A_{12}, A_{22})$, ..., $attcor_i(A_{1i}, A_{2i})$ is also a correspondence assertion, which states that $X_1\langle cor\rangle X_2$ is true. Moreover, for each $attcor_i(A_{1i}, A_{2i})$:

– if $attcor_i(A_{1i}, A_{2i})$ is $A_{1i} = A_{2i}$ then, at any time, for any $o \in (RWS(X_1) \cap RWS(X_2))$, $e_1.A_{1i} = e_2.A_{2i}$.

– if $attcor_i(A_{1i}, A_{2i})$ is $A_{1i} \supseteq A_{2i}$ then, at any time, for any $o \in (RWS(X_1) \cap RWS(X_2))$, $e_1.A_{1i} \supseteq e_2.A_{2i}$.

– if $attcor_i(A_{1i}, A_{2i})$ is $A_{1i} \cap A_{2i}$ then it is possible that for some $o \in (RWS(X_1) \cup RWS(X_2))$, $e_1.A_{1i} \supseteq e_2.A_{2i} \neq \emptyset$.

– if $attcor_i(A_{1i}, A_{2i})$ is $A_{1i} \neq A_{2i}$ then, at any time, for any $o \in (RWS(X_1) \cup RWS(X_2))$, $e_1.A_{1i} \supseteq e_2.A_{2i} = \emptyset$.

5 Information Source Integration

In this section we describe a technique that exploits sub-source similarities for constructing an integrated representation of information sources having different formats. The proposed technique uses the SDR-Network as the reference conceptual model for uniformly representing the information sources under consideration.

Our technique receives two information sources, represented by the corresponding SDR-Networks, and integrates them for obtaining a global SDR-Network SDR_G. To construct SDR_G, it carries out several activities. The first of them consists in the extraction of node synonymies, node homonymies and sub-source similarities relative to the SDR-Networks provided in input; to derive them, it is possible to exploit the approaches proposed in [Palopoli et al. 2001, Terracina and Ursino 2000].

After these properties have been extracted, the SDR-Networks under consideration are juxtaposed to obtain a (temporarily redundant and, possibly, ambiguous) global SDR-Network SDR_G. In order to normalize it, by removing its inconsistencies and ambiguities, several transformations must be carried out.

The first step of SDR_G normalization consists in deriving its root[1]. In particular, if the roots of the SDR-Networks in input are synonyms, they must be merged; otherwise, a new root is created and connected to them.

[1] Remember that SDR-Networks are rooted labeled graphs.

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

– *SDR-Network node examination.* First the Synonymy Dictionary SD and the Homonymy Dictionary HD are considered. For each tuple $\langle N_x, N_y, f_{xy} \rangle$ 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}. For each tuple $\langle N_x, N_y, f_{xy} \rangle$ 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.* The merge of nodes produces changes in the SDR_G topology; as a consequence, for each pair of nodes $[N_S, N_T]$ such that N_S derives from a merge process, it is necessary to check if N_S is connected to N_T by two arcs having the same direction[2] and, in the affirmative case, these arcs must be merged into a unique one. If only one arc exists from N_S to N_T, the corresponding semantic distance and relevance coefficients must be updated.

– *Sub-source examination.* This task exploits the Sub-source Similarity Dictionary SSD; in particular, for each tuple $\langle SS_x, SS_y, f_{xy} \rangle$ belonging to SSD, SS_x and SS_y must be "merged". The merge of sub-sources could lead to the presence of two arcs connecting the same pair of nodes and having the same direction; if this happens, the two arcs must be merged.

The complete algorithm for the integration of two information sources, represented by the corresponding SDR-Networks, is as follows:

Algorithm for the integration of two information sources
Input: a pair $SP = \{SDR_1, SDR_2\}$ of SDR-Networks;
Output: a global SDR-Network SDR_G;
var
    $Merged, NSet$: a set of SDR-Network nodes;
    AS: a set of SDR-Network arcs;
    N_x, N_y, N_{xy}, N_S, N_T, R_1, R_2: an SDR-Network node;
    A_1, A_2: an SDR-Network arc;
    SS_x, SS_y: a sub-source;
    SD: a Synonymy Dictionary;
    HD: a Homonymy Dictionary;
    SSD: a Sub-source Similarity Dictionary;
begin
    $[SD, HD, SSD] := Extract\_Interesting\_Properties(SP)$;
    $SDR_G := Juxtaposition(SP)$;

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

$R_1 := Get\_SDR\_Root(SDR_1)$;
$R_2 := Get\_SDR\_Root(SDR_2)$;
if $\langle R_1, R_2, f_{12}\rangle \notin SD$ **then**
    $Create\_Root(R_1, R_2, SDR_G)$;
$Merged := \emptyset$;
for each $\langle N_x, N_y, f_{xy}\rangle \in SD$ **do begin**
    $N_{xy} := Merge\_Nodes(N_x, N_y, SDR_G)$;
    $Merged := Merged \cup \{N_{xy}\}$;
end;
for each $\langle N_x, N_y, f_{xy}\rangle \in HD$ **do**
    $Rename\_Nodes(N_x, N_y)$;
$NSet := Get\_Nodes(SDR_G)$;
for each $N_S \in Merged$ **do**
    for each $N_T \in NSet$ **such that** $N_T \neq N_S$ **do begin**
        $AS := Get\_Arcs(N_S, N_T)$;
        if $(AS = \{A_1, A_2\})$ **then**
            $Merge\_Arcs(A_1, A_2, SDR_G)$;
        else if $(AS = \{A_1\})$ **then**
            $Update\_Coefficients(A_1, SDR_G)$;
    end;
for each $\langle SS_x, SS_y, f_{xy}\rangle \in SSD$ **such that**
$\langle Get\_Sub\text{-}source\_Root(SS_x), Get\_Sub\text{-}source\_Root(SS_y), g_{xy}\rangle \notin SD$ **do**
    $Merge\_Sub\text{-}sources(SS_x, SS_y, SDR_G)$;
for each $N_S \in NSet$ **do**
    for each $N_T \in NSet$ **such that** $N_T \neq N_S$ **do begin**
        $AS := Get\_Arcs(N_S, N_T)$;
        if $(AS = \{A_1, A_2\})$ **then**
            $Merge\_Arcs(A_1, A_2, SDR_G)$;
    end;
end

The procedure and the functions the algorithm activates have the following behaviour:

– *Extract_Interesting_Properties* receives a pair SP of SDR_Networks and derives the corresponding Synonymy Dictionary SD, Homonymy Dictionary HD and Sub-source Similarity Dictionary SSD. In order to obtain them, it implements the techniques described in [Palopoli et al. 2001, Terracina and Ursino 2000].

– *Juxtaposition* receives a pair SP of SDR-Networks and juxtaposes them for obtaining a (temporarily redundant and, possibly, ambiguous) global SDR-Network SDR_G.

– *Get_SDR_Root* takes an SDR-Network SDR_i as input and returns its root.

– *Create_Root* creates a root for SDR_G and links it to the roots of the two SDR-Networks composing SP.

- $Merge\_Nodes$ receives two nodes N_x and N_y and the global SDR-Network SDR_G and merges N_x and N_y for obtaining a unique node N_{xy}.
- *Rename_Nodes* receives two nodes N_x and N_y and renames at least one of them; it may require the support of the human domain expert for deciding which nodes must be renamed as well as the new names.
- *Get_Nodes* receives an SDR-Network SDR_i and returns the set of its nodes.
- *Get_Arcs* takes two nodes N_S and N_T as input and returns the set of arcs having N_S as the source node and N_T as the target node.
- $Merge\_Arcs$ receives two arcs A_1 and A_2 and a global SDR-Network SDR_G and merges A_1 and A_2 for obtaining a unique arc.
- *Update_Coefficients* receives an arc A and the corresponding SDR-Network SDR_G and updates the semantic distance and the semantic relevance coefficients associated with A.
- *Get_Sub-source_Root* receives a sub-source SS_i and returns its root.
- *Merge_Sub-sources* receives two sub-sources SS_x and SS_y, a global SDR-Network SDR_G and merges SS_x and SS_y for obtaining a unique sub-source.

The detailed description of this algorithm is beyond the scope of this paper; the interested reader can find it in [Rosaci et al. 2003].

6 An example case

In this section we show the peculiarities and the behaviour of CSC by means of an example case. In particular, we shall consider a mobile phone company called UNICOM. This is a big organization that collaborates with other big companies such as insurance and shipping firms, banks and so on; in this scenario, involved databases are probably heterogeneous in their semantics, representation formats and so on. As an example, some databases could be relational (i.e., structured data sources) whereas other ones could be XML documents (i.e., semi-structured information sources). Therefore, UNICOM is compelled to seek a solution for reconciling all these resources; CSC provides an answer to these exigencies by handling the interaction of structured and semi-structured data sources.

Consider three information sources relative to UNICOM. The first is an XML document and its DTD is shown in Figure 4. The second is an OEM graph and is shown in Figure 5. The third is a database and the corresponding relational schema is represented in Figure 6.

```
<!DOCTYPE UNICOM [
    <!ELEMENT Department EMPTY>
       <!ATTLIST Department
          <ID ID #REQUIRED
          <Name CDATA #REQUIRED
          <Building CDATA #REQUIRED
          <Floor CDATA #REQUIRED
          <Off_Incharge CDATA #REQUIRED
       >
    <!ELEMENT Cell_Phone (Data?, Brand?, Cellnum?)>
       <!ATTLIST Cell_Phone
          <ID ID #REQUIRED
          <Holder IDREFS #IMPLIED
          <Frequence IDREFS #IMPLIED
       >
    <!ELEMENT Date (#PCDATA)>
    <!ELEMENT Cellnum (#PCDATA)>
    <!ELEMENT Brand (#PCDATA)>
    <!ELEMENT Client EMPTY>
       <!ATTLIST Client
          <ID ID #REQUIRED
          <Cellnum CDATA #REQUIRED
          <Name CDATA #REQUIRED
          <DoB CDATA #REQUIRED
          <Address CDATA #REQUIRED
          <National_Id CDATA #REQUIRED
          <Purchase_Date CDATA #IMPLIED
       >
]>
```

Figure 4: The DTD of an XML document relative to UNICOM

The extracted source data are passed to the TIM component. The human expert supplies the correspondence assertions to equip the component with inter-agency capability. Then, the module reconciles the semantics from schemas and produces the mappings. These are forwarded to the RMA-Extractor where the information is filtered in order to get the relevant data. The human expert again supplies some parameters that facilitate the filtering. Filtered data are delivered to the RMA Graph Builder where the global SDR-Network is constructed. This is forwarded to the Query Formulator where the human expert provides also some object patterns and references that support the formulation of global queries. Each global query is decomposed into a set of component queries; each of these is, then, executed and the obtained answers are, then, composed for constructing a global answer.

Example 4. Consider the information sources:

IS_2=MOBILE_PHONE(brand,model,size,price)

and

IS_3=M_PHONE(brand,model,measurement,frequency,cost)

```
UNICOM: &1
    {Department: &2
       {Number: &5 ‘‘Cell05’’,
       Name: &6 ‘‘Accounts’’,
       Building: &7 19,
       Floor: &8 7,
          Off_Incharge: &9,
             {Lastname: &18 ‘‘Mbale’’,
             Firstname &19 ‘‘Jameson’’}}},
    {Cell_Phone: &3
       {ID: &10 0505630,
       Name: &11 ‘‘Nokiya’’,
       Frequency: &12 ‘‘GSM900/1800’’}},
    {Client: &4
       {ID: &13 19690302,
       Name: &14 ‘‘Barbara’’,
       Address: &15,
          {Street: &21 ‘‘Xidazhi’’,
          AreaCode: &22 15001,
          District: &23 ‘‘Nangang’’,
          City: &24 ‘‘Harbin’’},
       Cellnum: &16 13089984242,
       DoB: &17 ‘‘Feb. 03 1969’’}}
```

Figure 5: An OEM Graph relative to UNICOM

```
Department(Number, Name, Building, Floor);
Cell_Phone(ID, Name, Frequency);
Client(ID, Name, Address, Cellnum, DoB).
```

Figure 6: A database relative to UNICOM

Assume these sources have been elaborated by CSC and object patterns have been obtained. Assume, now, the following global query is formulated to retrieve data on IS_2 and IS_3:

GQ_1 = SELECT model, size, cost
WHERE brand LIKE “Nokiya”

The Query Formulator passes GQ_1 to the Query Decomposer. This decomposes GQ_1 into the queries:

$GQ_1^{IS_2}$ = SELECT IS_2.model, IS_2.size, IS_2.price
WHERE IS_2.brand LIKE “Nokiya”

$GQ_1^{IS_3}$ = SELECT IS_3.model, IS_2.measurement, IS_2.cost
WHERE IS_2.brand LIKE “Nokiya”

$GQ_1^{IS_2}$ and $GQ_1^{IS_3}$ are, then, executed and the corresponding answers are collected by the Query Composer that produces the global answer and sends it to the Query Formulator.

7 Analysis of CSC

The necessity of integrating structured and semi-structured data sources has led researchers to design complicated architectures that are not easily handled. CSC is an attempt to construct a less complicated architecture that uses correspondence assertions to make the system capable of capturing the semantics of heterogeneous data sources. In addition, since it is based on the SDR-Network conceptual model, it is capable of managing different data representation formats such as XML documents, OEM Graphs and E/R schemas; this capability is quite difficult to find in other systems already proposed in the literature.

Another interesting feature of CSC is its ability to allow the expert to design the processed information towards the user applications' requirements. This is a particularly interesting advantage since it allows time and money in not dealing with unnecessary processes of bulk data to be saved.

In addition, the use of facilitators, such as correspondence assertions, allows the system to be dynamic and to be exploited in various application fields such as insurance, medicine, engineering, mining, education, Web and so on.

Thanks to the exploitation of the global schema, CSC allows the user to support the formulation of generic instead of a pre-defined set of queries.

Finally, differently from other systems already proposed in the literature, CSC allows the formulation of queries even if the structure of the original information sources is not known.

8 Conclusions and future work

In this paper we have investigated an approach for allowing the integration of both structured and semi-structured data. More specifically, we have introduced a system, named CSC, capable of guaranteeing such a feature. We have seen that CSC exploits correspondence assertions for equipping the TIM component with enough intelligence to handle the correspondence between constructs of different models. The exploitation of the SDR-Network conceptual model allows CSC to handle information sources having different data representation formats such as XML documents, OEM Graphs and E/R schemas.

We have seen also that CSC is provided with components allowing the formulation of global queries, the decomposition of each of them into sub-queries, the execution of these last and the composition of the returned answers in such a way as to obtain a global answer.

The presence of all these features allows us to claim that CSC is capable of managing the integration of both structured and semi-structured data.

Presently, CSC requires human intervention on three occasions. As for our research efforts, we plan to investigate how to reduce the occasions in which human intervention is necessary.

References

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

[Abiteboul et al. 1997] S. Abiteboul, D. Quass, J. McHugh, J. Widom, and J.L. Wiener. The lorel query language for semistructured data. *International Journal on Digital Libraries*, 1(1):68–88, 1997.

[Arens et al. 1993] Y. Arens, C. Y. Chee, C. Hsu, and C. A. Knoblock. Retrieving and integrating data from multiple information sources. *International Journal of Cooperative Information Systems*, 2(2):127–158, 1993.

[Beneventano et al. 1997] D. Beneventano, S. Bergamaschi, C. Sartori, and M. Vincini. ODB-Tools: A description logics based tool for schema validation and semantic query optimization in object oriented databases. In *Proc. of Advances in Artificial Intelligence, 5th Congress of the Italian Association for Artificial Intelligence (AI*IA'97)*, pages 435–438, Roma, Italy, 1997. Lecture Notes in Computer Science, Springer Verlag.

[Buneman et al. 1997] P. Buneman, S. Davidson, M. Fernandez, and D. Suciu. Adding structure to unstructured data. In *Proc. of International Conference on Database Theory (ICDT'97)*, pages 336–350, Delphi, Greece, 1997. Lecture Notes in Computer Science, Springer-Verlag.

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

[Castano et al. 2002] S. Castano, V. De Antonellis, A. Ferrara, and G. Kuruvilla. Ontology-based integration of heterogeneous XML datasources. In *Atti del Decimo Convegno Nazionale su Sistemi Evoluti per Basi di Dati(SEBD'02)*, pages 27–41, Portoferraio, Italy, 2002.

[Doan et al. 2003] A. Doan, P. Domingos, and A. Halevy. Learning to match the schemas of databases: A multistrategy approach. *Machine Learning Journal*, 50:279–301, 2003.

[Garcia-Molina 1997] H. Garcia-Molina, Y. Papakonstantinou, D. Quass, A. Rajaraman, Y. Sagiv, J. Ullman, V. Vassalos, and J. Widom. The TSIMMIS approach to mediation: Data models and languages. *Journal of Intelligent Information Systems*, 8:117–132, 1997.

[Goldman and Widom 1997] R. Goldman and J. Widom. Dataguides: Enabling query formulation and optimization in semistructured databases. In *Proc. of Very Large Data Bases (VLDB'97)*, pages 436–445, Athens, Greece, 1997. Morgan Kaufman.

[Konopnicki and Shumueli 1995] D. Konopnicki and O. Shumueli. W3QS: A query system for the world wide web. In *Proc. of International Conference on Very Large Data Bases (VLDB '95)*, pages 54–65. Morgan Kaufman, 1995.

[Konopnicki and Shumueli 1997] D. Konopnicki and O. Shumueli. W3QS: A system for WWW querying. In *Proc. of International Conference on Very Large Data Bases (VLDB '97)*, page 586, Birmingham, England, UK, 1997. IEEE Computer Society.

[Larson et al. 1989] J.A. Larson, S.B. Navathe, and R. Elmastri. A theory of attribute equivalence in databases with application to schema integration. *IEEE Transactions on Software Engineering*, 15(4):449–463, 1989.

[Li and Clifton 2000] W. Li and C. Clifton. SEMINT: A tool for identifying attribute correspondences in heterogeneous databases using neural networks. *Data and Knowledge Engineering*, 33(1):49–84, 2000.

[Lim and Ng 2001] S. Lim and Y. Ng. Semantic integration of semistructured data. In *Proc. of Third International Symposium on Cooperative Database Systems and Applications (CODAS'01)*, pages 15–24, Beijing, China, 2001. IEEE Computer Society Press.

[Madhavan et al. 2001] J. Madhavan, P.A. Bernstein, and E. Rahm. Generic schema matching with cupid. In *Proc. of International Conference on Very Large Data Bases (VLDB 2001)*, pages 49–58, Roma, Italy, 2001. Morgan Kaufmann.

[McBrien and Poulovassilis 2001] P. McBrien and A. Poulovassilis. A semantic approach to integrating XML and structured data sources. In *Proc. of 13th Conference on Advanced Information Systems Engineering (CAiSE'01)*, pages 330–345, Interlaken, Switzerland, 2001. Lecture Notes in Computer Science, Springer-Verlag.

[Mendelzon et al. 1996] A. O. Mendelzon, G. A. Mihaila, and T. Milo. Querying the world wide web. In *Proc. of Conference on Parallel and Distributed Information Systems (PDIS'96)*, pages 80–91, Miami Beach (Florida), USA, 1996. IEEE Computer Society.

[Milo and Zohar 1998] T. Milo and S. Zohar. Using schema matching to simplify heterogenous data translations. In *Proc. of International Conference on Very Large Data Bases (VLDB'98)*, pages 122–133, New York City, USA, 1998. Morgan Kaufmann.

[Nestorov et al. 1998] S. Nestorov, S. Abiteboul, and R. Motwani. Extracting schema from semistructured data. In *Proc. of International Conference on Management of Data (SIGMOD'98)*, pages 295–306, Seattle, Washington, USA, 1998. ACM Press.

[Ouksel and Naiman 1994] A.M. Ouksel and C.F. Naiman. Coordinating context building in heterogeneous information systems. *Journal of Intelligent Information Systems*, 3(2):151–183, 1994.

[Palopoli et al. 2001] 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 330–337, Heidelberg, Germany, 2001. IEEE Computer Society.

[Parent and Spaccapietra 1995] C. Parent and S. Spaccapietra. Database integration: An overview of issues and approaches. *Communication of the ACM*, 41(5), 1998.

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

[Rosaci et al. 2003] D. Rosaci, G. Terracina, and D. Ursino. An approach for deriving a global representation of data sources having different formats and structures. *Knowledge and Information Systems*, Forthcoming.

[Roth and Schwarz 1997] M.T. Roth and P.M. Schwarz. Don't scrap it, wrap it! a wrapper architecture for legacy data sources. In *Proc. of International Conference on Very Large Data Bases (VLDB 1997)*, pages 266–275, Athens, Greece, 1997. Morgan Kaufmann.

[Scheuermann et al. 1998] P. Scheuermann, W. Li, and C. Clifton. Multidatabase query processing with uncertainty in global keys and attribute values. *Journal of the American Society for Information Science*, 49(3):283–301, 1998.

[Terracina and Ursino 2000] 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, Pune, India, 2000. McGraw Hill.