Schema Matching in the Semantic Web Environment

Linková, Zdeňka
2007

Schema Matching in the Semantic Web Environment

Post-Graduate Student:
ING. ZDEŇKA LINKOVÁ

Department of Mathematics
Faculty of Nuclear Science and Physical Engineering
Czech Technical University
Trojanova 13
120 00 Prague 2, CZ

Institute of Computer Science of the ASCR, v. v. i.
Pod Vodárenskou věží 2

182 07 Prague 8, CZ

linkova@cs.cas.cz

Supervisor:
ING. JÚLIUS ŠTULLER, CSC.

Institute of Computer Science of the ASCR, v. v. i.
Pod Vodárenskou věží 2

182 07 Prague 8, CZ

stuller@cs.cas.cz

Field of Study:
Mathematical Engineering

The work was supported by the project 1ET100300419 of the Program Information Society (of the Thematic Program II of the National Research Program of the Czech Republic) “Intelligent Models, Algorithms, Methods and Tools for the Semantic Web Realization” and partly by the Institutional Research Plan AV0Z10300504 “Computer Science for the Information Society: Models, Algorithms, Applications”.

Abstract

The paper deals with one step of the non-materialized data integration - the schema matching task. The proposal is for data sources on the Semantic Web; the crucial assumption for the considered task is the availability of the ontologies describing data to integrate. These ontologies are used to find correspondences between source schemas elements. For this, the techniques from ontology alignment and ontology merging field are used.

1. Introduction

Data integration [1] is a task of combining data residing at different sources and enabling a user to process these data as a one whole. When data integration is non-materialized [2], the issue usually is to provide an unified view over the data sources. This view is then accessed as a new - integrated - data source containing all the data. However, in fact, the view is only virtual and does not store any data; the data physically stay stored in the original sources. In order to enable accessing them through the integrated view, connections between the schema of such an “integrated” view and the schemas of the data sources have to be established.

The integration process can be seen as a collection of several tasks, which together brings the required result. The basic steps of such a data integration are following:

- **Schema matching** - Under an assumption the data sources to integrate have been constructed independently, and their schemas were designed by different designers for different purposes, the data schemas are in general heterogeneous. Therefore, it is crucial, for their processing together, to find correspondences between them. The problem of *finding* schemas correspondences is called schema matching [4], [5].

- **Schema mapping** - A usual way to *describe* the correspondences between schemas of the integrated data sources is to use mappings.
  A mapping can be seen as a structure, e.g. a set of assertions, that establishes a connection between elements of the view schema (usually called global schema) and the data source schemas (local schemas).
  Two basic approaches [6], [7] have been used in order to specify the mappings: *a Global As View* (GAV) approach consisting in defining the global schema as a set of views over the local schemas and *a Local As View* (LAV) approach consisting in defining the local sources as a set of views made on the global schema.

- **Query processing** - The composition of mapping is an essential task. It plays a crucial role in querying - another important process of a data integration.
  Within a data integration (system), a user poses his query on the global view in terms of the glo-

bal view. In order to execute the query over the sources, where data are stored, query processing [3] is needed.
There are two approaches to query processing. The first one is *query rewriting* - a query is decomposed to parts referring to local sources and reformulated to be expressed in local source background. The other one is *query answering* - it do not pose any limitations on how a query is processed, the only goal is to exploit all possible information to compute the answer, for example find the set of data such that the knowledge represented in the data logically implies that it is an answer to the query.

This paper deals with the first step of the integration process. It considers web data sources. In general, data of the current World Wide Web are distributed in many sources, having different formats, heterogeneous (or none) schemas, etc., and so, processing of them is very difficult. Therefore, the data taken into consideration in this approach are restricted to the Semantic Web environment.

Semantic Web [8], [9], [10] is intended as a semantic extension of the current web. Nowadays, the main techniques for the Semantic Web data description are:

- XML language [11] for the data structuring
- RDF(S) [12], [13] for the metadata description
- OWL [14] for the ontology specification

Taking only the Semantic Web data into consideration, data sources with defined ontologies [15] represent a natural assumption for the approach.

The paper is organized as follows: Section 2 introduces the matching operation and brings a brief review of schema matching approaches. Section 3 is concerned with ontology-based schema matching approach; it deals not only with matching discovery, but also with expression of found mappings. Finally, Section 4 summarizes the paper.

2. Schema Matching

The matching operation takes two schemas as an input and produces as an output mappings describing schemas relationships. The task of finding corresponding mappings is a topic of many research projects. Unfortunately, in many projects and implementations, it has been solved mainly manually [16]. This has significant limitations - it is time consuming, prone to errors and expensive. An effort to automatize this as much as possible resulted at most in providing the so-called match candidates, and the user needs to (manually) adjust the assignment to guarantee their suitability. This is because the schemas have very often some not expressed semantics that affects the matching.

Many ways how to search for schema correspondences where investigated in the past. The approaches can be basically distinguished according to the information level, at which the schemas have been compared:

- **Instance** - At an instance level, matching approaches consider instance data to find the correspondences between the schemas describing these data.
- **Term** - Approaches working at a term level are linguistic-based (e.g., based on names and textual descriptions of schema elements). They can work with terms relations (synonyms, homonyms, etc.) or can be string-based (considering used terms as a character string and comparing them in order to find their relations as prefix, suffix, root, etc.).
- **Structure** - Matching can be performed for individual schema elements, such as attributes, or for combinations of elements, such as complex schema structures. At this level, for instance, graph-based techniques are used.

For example, approaches comparing particular schema attributes can be based on their names (optionally taking into account also known synonym relationships or using lexical techniques), data types, active domains; some of them deal also with the structures of the sources.

A matching possibility obtained by this is often expressed using some similarity function. This similarity can be based on probability [17], on the cosine measure of particular attribute feature vectors [18], or some other measure describing the number of explored aspects in which they correspond [19]. These measures can be further used for selecting matching from found candidates. Sometimes, some additional techniques like candidates refinement [20] or machine learning [21] are used.

3. Ontology-based Schema Matching

In the proposed approach to schema matching, available ontologies describing data in the integrated sources are supposed. From them, required correspondences between particular schema elements will be derived.

Generally, an element can participate in zero, one or many correspondences searched within the matching data schemas. Moreover, an individual element (of some schema) can match one or more elements (of another schema). Therefore, also a term *matching cardinality* is usually used. With respect to a mapping element, matching can be of a cardinality 1:1, 1:N, N:1, N:M. Most existing approaches match each element of one schema to the element of another schema with cardinality 1:1 or 1:N.

This approach considers correspondences of cardinality:

- **1:1** when matching two schemas. This means that one element of the first schema is matched to one element of the other schema.
- **1:N** when matching one schema to more schemas. This can be seen as a set of matching used in the case above. Mentioned 1:N matching is often used in data integration for matching a schema of a global virtual view and schemas of local data sources.

To formalize the notion of the required matching correspondences, a matching of a cardinality 1:1 is an assertion:

$$\varepsilon_1 \ \ \rho \ \ \varepsilon_2$$

where

ε_1 is an element of one schema
ε_2 is an element of the other schema
ρ is a relation between ε_1 and ε_2 expressing their correspondence.

A matching of a cardinality 1:N is a set of assertions of 1:1 cardinality:

$$\{\varepsilon_1 \ \ \rho_i \ \ \varepsilon_i\}$$

where

ε_1 is an element of one schema
ε_i is an element of another schema
ρ_i is a relation between ε_1 and ε_i expressing their correspondence.

The relation ρ can be one of the following kinds of correspondence:

- **Is-a** hierarchical relationship (i.e. one element is more general than the other or vice versa).
  This kind is denoted as $\supseteq$, respective $\subseteq$.
- **Equivalence** between the elements.
  This kind is denoted as =.
- **Disjointness**, i.e. elements cannot be matched in any way.

3.1. Schema Mapping

A result of the matching task, found schema correspondences, is often called *schema mapping*. In general, for schema mapping, an arbitrary structure can be used. Mapping can be done in a broad scale from the simplest *one-to-one mapping rules* expressing direct correspondences between elements, trough *mapping a concept to a query or a view* [22] (e.g. respecting GAV or LAV approach), to some additional mapping structures (e.g. a reference model in [23]). Different projects usually use their own notion of mapping.

However, instead of using for instance mapping rules as assertions for global and local schemas elements that are particular approach oriented, a more complex and even standardized structure covering all mapping can be employed. An *OWL ontology* will be used to describe the mapping between elements of the global view and the local sources.

The use of an ontology for the mapping brings a possibility to reuse it in other tasks or situations. Also, when deriving further correspondences, taking another schema (of another data source) into account for instance, mapping described in an ontology can be seen as another ontology available for compared sources. Moreover, for the future, considering also other kinds of correspondences, an ontology can be employed, because it can capture various relation types.

To capture the mapping, according to the type of the matching, an appropriate OWL [14] construct is used. In OWL, classes provide an abstraction mechanism for grouping described resources. On the Web, resource is every thing or entity that can be identified. A notion of `owl:Class` is therefore used for elements correspondences:

- For the **is-a** relationship, i.e. $\varepsilon_1 \subseteq \varepsilon_2$, the notion of subclass can be employed.
  An appropriate OWL feature for this is `rdfs:subClassOf`, which allows one to say that an extension of a class description is a subset of an extension of another class description.
- For the **equivalence** relationship, i.e. $\varepsilon_1 = \varepsilon_2$, an OWL feature `owl:equivalentClass` can be used.
  `owl:equivalentClass` allows one to say that a class description has exactly the same class extension as another class description.

However, also in this case `rdfs:subClassOf` can be used: defining ε_1 as subclass of ε_2 and at the same time ε_2 as subclass of ε_1, it is possible to say that ε_1 and ε_2 are equivalent classes.

- The **disjointness** (i.e. to say that an extension of a class description has no members in common with an extension of another class description) can be expressed by `owl:disjointWith`.

3.2. Matching with Shared Ontology Available

In the simplest case, a description of all the sources is covered by the only one ontology. This ontology is shared by the sources and captures all the data description. Schema elements correspondences can be directly find in the given ontology.

For this, following assumption is adopted:

The semantic relationship between terms defined in the ontology implies the same relationship between schema elements labeled by these terms.

Considering previously stated correspondences types class-subclass and class equivalence, an is-a hierarchy defined by the shared ontology is used. When matching two data source schemas, for each element of the first schema and for each element of the other one, their relationship is searched in the ontology - if an is-a relationship is defined in the ontology, the appropriate correspondence is between the compared elements.

Some relationships need not be in the ontology directly expressed, however, they can be obtained using transitivity of subclass relationship. For example, when approaching an ontology as a graph with classes as nodes and is-a relationship labeled edges, found correspondence between two elements means not only an existing edge of that label, but also an oriented path between the classes appropriately labeled.

When classes are disjoint, it means that there should not be any is-a hierarchy relationship between them, and, therefore, it is not needed to search it. However, this situation leads in practice to the same effect as relationships had been searched, but none has been found.

As, to capture the mapping, an OWL expression is used, the given shared ontology can be seen as a "superontology" of the searched mapping, in that sense that it describes all the classes and their relationships as stated in the mapping.

Note that all correspondences derived from the given ontology are adopted; they are not considered only as matching candidates, because there is no correspondence estimation - all of them are defined in the ontology. This step demand no (human) user interaction.

3.3. General Matching with Ontologies

Generally, for definitions of terms in the sources, more ontologies are used. Some sources can use for some terms a shared ontology, but it does need to cover all the terms, and the use of a shared ontology cannot be assumed. Instead of it, all supported ontologies have to be considered.

By merging all given ontologies, a “new” shared ontology is obtained, and this general case can be transformed to the previous one. For doing this, ontology alignment or ontology merging methods can be employed.

In the context of ontologies, terms alignment and merging are closely related [24]. For both, also matching and mapping are relevant. *Ontology alignment* usually means a task of establishing a collection of binary relations between two ontologies. This allows to define a way for merging of ontologies. *Ontology merging* results in a new, integrated ontology.

Methods for *matching in the field of ontology merging* or *ontology alignment* are of *similar principles* to the *methods for schema matching*. That is, because ontologies and data schemas are closely related. The main difference is a purpose. An ontology is developed in order to define a meaning of terms used in some domain, whereas a schema is developed in order to model some particular data. Especially for schemas using a semantic data model, there is often no obvious difference and way to identify which representation is a schema and which is an ontology. In practise, schemas and ontologies usually have both well defined terms and contexts of their occurrence. Because data schemas often do not provide explicit semantics for their data, matching is usually performed with the help of techniques trying to guess the meaning of used terms. When assuming available data source ontologies, this is not needed.

Methods for ontology alignment or ontology merging are performed, as methods for schema matching, at different levels: *instance* (e.g. comparing set of instances), *element* (e.g. lexical techniques), and *structure* (e.g. graphs techniques), and use syntactic and semantic approach.

Also similarity with so-called match candidates can be found. Methods therefore require user interaction or use some heuristics based on user earlier decisions. Note, although in the case of a shared ontology, there are no candidates, and correspondences are strictly derived, the candidates can arise from this subtask.

Ontology merging methods are topics of many research projects:

- **Chimaera** [25] - The Chimaera system tool provides *support for merging of different ontologies* that may have been written by different authors using different vocabularies. It is based on a Ontolingua ontology editor [26], and considers only the class-subclass relation.

  Chimaera is an interactive merging tool that demand user interaction: it generates name resolution lists that help the user in the merging task by suggesting terms each of which is from a different ontology that are candidates to be merged or to have taxonomic relationships not yet included in the merged ontology. Chimaera leaves the decision of what to do entirely to the user and does not make any suggestions itself.

- **PROMPT** [27] - The PROMPT is an algorithm for *semi-automatic ontology merging and alignment*. It performs some tasks automatically and guides the user in performing other tasks for which his intervention is required. It also determines possible inconsistencies, which result from user actions, and suggests ways to resolve these inconsistencies.

  First, PROMPT creates an initial list of matches based on class names. Then follows the cycle of selecting candidates (by the user) and automatically executed actions - the algorithm works with data types, considers linguistically similar names and subclass hierarchy.

  The PROMPT ontology merging algorithm was implemented as an *extension to the Protégé-2000* [28] ontology editor.

- **FCA-MERGE** [29] - The FCA-MERGE is a method for merging ontologies following a *bottom-up approach* which offers a *structural description of the merging process*.
  For the source ontologies, it extracts instances from a given set of domain-specific text documents relevant to the merged ontologies by applying natural language processing techniques.

  Based on the extracted instances, mathematically founded techniques taken from Formal Concept Analysis [30] are applied to derive a lattice of concepts as a structural result of FCA-MERGE. Instance extraction and the FCA-MERGE core algorithm are fully automatic. The generated result is then explored and transformed into the merged ontology with human interaction.

- **HCONE** [31] - HCONE approach on ontology merging *exploits WordNet* [32], which is an external natural language information source. The HCONE method consults WordNet for lexical information. Linguistic and structural knowledge about ontologies are exploited by the Latent Semantics Indexing method (LSI - a vector space technique for information retrieval and indexing) [33] for associating concepts to their informal, human-oriented intended interpretations realized by WordNet senses.

  Using concept intended semantics, the proposed method translates formal concept definitions to a common vocabulary and exploits the translated definitions by means of description logics reasoning services. The goal is to validate the mapping between ontologies and to find a minimum set of axioms for the merged ontology. The HCONE approach is not completely automated; human involvement is placed at the early stages of the mapping/ merging process.

4. Summary and Conclusion

Schema matching is a crucial part of a data integration process. The matching result, a mapping, is then used when accessing integrated data. For schema matching, several techniques based on various information about data sources are employed. With source ontologies available, it is possible to derive the requested correspondences between data schemas.

An important issue is a way how to express the found mapping. In this approach, an OWL ontology is used. This brings a possibility to share or reuse the derived mapping. The mapping expressed in a standardized way can be further used in other situations and accessed also by various tools. In particular, this mapping allows to use techniques developed for ontology processing.

If an ontology shared by all the data sources is supported, mapping of source schemas can be easily obtained from this ontology. Generally, if there are two or more ontologies used for data description, these ontologies are merged. Ontology merging results in getting a shared otology as stated earlier, and the mapping can be then obtained. So, by this approach, the task of schema matching is transformed to the ontology merging task, for which, there are available methods and tools that can be employed.

An ontology-based schema matching is a subtask of ontology-based data integration which will be studied more in my thesis.

References

[1] Z. Bellahsene, "Data integration over the Web", *Data & Knowledge Engineering*, 44 (2003), pp. 265-266.

[2] J. D. Ullman, "Information integration using logical views", *Theoretical Computer Science* 239 (2000), pp. 189-210.

[3] R. Pottinger and A. Levy, "A Scalable Algorithm for Answering Queries Using Views", In the Proceedings of the *26th VLDB Conference*, Cairo, Egypt (2000).

[4] E. Rahm and P. A. Bernstein, "A survey of approaches to automatic schema matching", In *VLDB Journal: Very Large Data Bases*, 10(4), pp. 334-350, 2001.

[5] P. Shvaiko and J. Euzenat, "A survey of schema-based matching approaches", 3730, pp. 146-171, 2005.

[6] M. Lenzerini, "Data Integration: A Theoretical Perspective", In Proceedings of the *21st ACM SIGMOD - SIGACT - SIGART symposium on Principles of database systems*, ACM Press, pp. 233-246, 2002.

[7] A. Cali, D. Calvanese, G. De Giacomo, and M. Lenzerini, "On the Expressive Power of Data Integration Systems", In Proceedings of the *21st Int. Conf. On Conceptual Modeling (ER 2002)*, LNCS 2503, Springer, pp. 338-350, 2002.

[8] T. Berners-Lee, J. Hendler and O. Lassila, "The Semantic Web", *Scientific American*, vol. 284, 5, pp. 35-43, 2001.

[9] M.-R. Koivunen and E. Miller, "W3C Semantic Web Activity", in the proceedings of the *Semantic Web Kick/off Seminar*, Finland, 2001.

[10] J. Euzenat. "Research challenges and perspectives of the Semantic Web". Report of the EU-NSF Strategic Research Workshop, Sophia-Antipolis, France, October, 2001.

[11] Extensible Markup Language (XML), `http://www.w3.org/XML/`.

[12] Resource Description Framework (RDF), `http://www.w3.org/RDF/`.

[13] RDF Vocabulary Description Language 1.0: RDF Schema, *W3C Recommendation*, `http://www.w3.org/TR/2004/REC-rdf-schema-20040210`.

[14] Web Ontology Language (OWL), `http://www.w3.org/2004/OWL`.

[15] Y. Ding, D. Fensel, M. Klein, and B. Omelayenko, "The semantic web: yet another hip?", *Data & Knowledge Engineering*, 41 (2002), pp. 205-227.

[16] P. Mitra, G. Wiederhold, and J. Jannink, "Semi-automatic integration of knowledge sources", In Proceeding of the *2nd Int. Conf. On Information FUSION'99*, 1999.

[17] H. Nottelmann and U. Straccia, "Information retrieval and machine learning for probabilistic schema matching", *Inf. Process. Manage.*, 43(3), pp. 552-576, 2007.

[18] X. Su and J. A. Gulla, "An information retrieval approach to ontology mapping", *Data & Knowledge Engineering* 58(1), pp. 47-69, 2006.

[19] S. Yi, B. Huang, and W. T. Chan, "Xml application schema matching using similarity measure and relaxation labeling", *Inf. Sci.*, 169(1-2), pp. 27-46, 2005.

[20] H.-H. Doa and E. Rahmb, "Matching large schemas: Approaches and evaluation", *Information Systems*, (in print), 2007.

[21] L. Xu and D. W. Embley, "A composite approach to automating direct and indirect schema mappings", *Inf. Syst.*, 31(8), pp. 697-732, 2006.

[22] D. Calvanese, G. De Giacomo, and M. Lenzerini, "Ontology of integration and integration of ontologies", In Proceedings of the *2001 Description Logic Workshop (DL 2001)*, 2001.

[23] H. T. Uitermark, P. J. M. van Oosterom, N. J. I. Mars, and M. Molenaar, "Ontology-based integration of topographic data sets", *International Journal of Applied Earth Observation and Geoinformation* 7 (2005), pp. 97-106.

[24] Y. Kalfoglou and M. Schorlemmer, "Ontology mapping: the state of the art", *The Knowledge Engineering Review* 18(1), pp. 1-31, 2003.

[25] D. L. McGuinness, R. Fikes, J. Rice, and S. Wilder, "An Environment for Merging and Testing Large Ontologies", In Proceedings of the *Seventh International Conference*, 2000.

[26] A. Farquhar, R. Fikes, and J. Rice, "The Ontolingua Server: a Tool for Collaborative Ontology Construction", Technical report, Stanford KSL 96-26, 1996.

[27] N. F. Noy and M. A. Musen, "PROMPT: Algorithm and Tool for Automated Ontology Merging and Alignment", In *AAAI/IAAI*, pp. 450-455, 2000.

[28] The Protégé Ontology Editor and Knowledge Acquisition System, `http://protege.stanford.edu/`.

[29] G. Stumme and A. Maedche "FCA-MERGE: Bottom-Up Merging of Ontologies", In *IJCAI*, pp. 225-234, 2001.

[30] U. Priss, "Formal Concept Analysis in Information Science (draft)", `http://www.upriss.org.uk/papers/arist.pdf`.

[31] K. Kotis and G. A. Vouros, "The HCONE Approach to Ontology Merging", In *ESWS*, LNCS 3053, Springer, pp. 137-151, 2004.

[32] WordNet, a lexical database for the English language, `http://wordnet.princeton.edu/`.

[33] S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman, "Indexing by Latent Semantic Analysis", *Journal of the American Society of Information Science* 41(6), pp. 391-407, 1990.