Ontology-Based Schema Integration

Zdeňka Linková

Institute of Computer Science, Academy of Sciences of the Czech Republic
Pod Vodárenskou věží 2, 182 07 Prague 8, Czech Republic
linkova@cs.cas.cz
Department of mathematics, Faculty of nuclear science and physical engineering,
Czech Technical University in Prague
Trojanova 13, 120 00 Prague 2, Czech Republic

Abstract. Data integration usually provides a unified global view over several data sources. A crucial part of the task is the establishment of the connection between the global view and the local sources. For this purpose, two basic mapping approaches have been proposed: GAV (Global As View) and LAV (Local As View). On the Semantic Web, there can be considered also an ontological approach.
In this paper, data integration is solved using ontologies of the sources. To express relationships between the global view and local source schemas, an ontology for the integration system is built. Thus, a schema integration task is transformed to an ontology merging task.

1 Introduction

Today's world is a world of information. The expansion of World Wide Web has brought a number of information sources. However, at the same time, a number of different formats, data heterogeneity, and not yet efficient machine processing of web sources cause many problems. One of them is the reappeared problem of data integration.

Data integration is the task of combining data residing at different sources and enabling the user to process these data as one whole. Data integration has been an acknowledged data processing problem for a long time. Although there have been some projects on integration of data within particular areas, there is no universal tool for general data integration.

In general, data integration can be pursued in different layers. It is possible to consider only data, or consider also metadata (e.g. schemas). With greater data amount, the integration approach is rather non-materialized than materialized. The integration result brings virtual view over data sources that do not store any data. Therefor, the establishment of the connection to original data sources is crucial. To consider the data schemas is essential. There are some basic approaches to the design a non-materialized integration system, each with its advantages and disadvantages. The proposed approach brings an idea from the Semantic Web - a semantic extension of the current World Wide Web.

The paper is organized as follows. Section 2 describes the data integration task and basic approaches. Section 3 is concerned with related ontology-based data integration approaches. Section 4 presents an ontological approach to data integration. Finally, Section 5 summarizes the paper.

2 Data integration

In data integration, the goal is to synthesize data from different data sources into one integrated data source. A user willing to process the data uses the integrated source and is freed from the knowledge where the data are and how the data are structured in the respective sources.

The integrated data source can be materialized, i. e. a new data source is created and it physically stores data, or it can be virtual, i. e. a virtual view is defined and the data remain in the sources. In materialized data integration approach, a copy of the data is made. So, with respect to actualization requirements, it is suitable for more or less stable data. Virtual data integration approach provide an interface to autonomous data sources, it can be used also for large amount of data with relatively frequently changing content. In a connection with the World Wide Web data, this approach suggests itself. It is also possible to combine both approaches. An example is an integration system that provides a virtual integrated view, but it also materializes some data in a cache. The cache is usually used for frequently accessed data.

A commonly used system architecture in virtual view approach [1] to data integration is depicted in Figure 1. A base of the system is a set of data sources to integrate. The higher layer is a set of components called wrappers. Each wrapper belongs to one local data source and it plays a role of a connector between the local source background (it means a specific model, a specific language etc. for the source) and the global one. The pure integration part of the system is presented in hierarchical layers of mediator components. A mediator can obtain information from components below it and can provide information to components above it. In general, an integration system can contain an arbitrary complex architecture.

Each mediator in a hierarchy can be seen as a virtual view. These views are then used in query evaluation. A user of the integration system poses his query to a global view using a global schema. Using mediation integration, the query is reformulated and decomposed to refer to the data sources and the queries are also executed over the sources. Then obtained information is composed and the answer is given back to the user.

The main components of a data integration system are the sources with their local schemas, the global virtual view with the global schema, and the mediated system that expresses the correspondence between the global source and the local sources. So it follows that a data integration system I can be seen [2] as a triple

$$I = (G, L, M),$$

where G is a global schema, L is a set of local schemas and M is a mediation system.

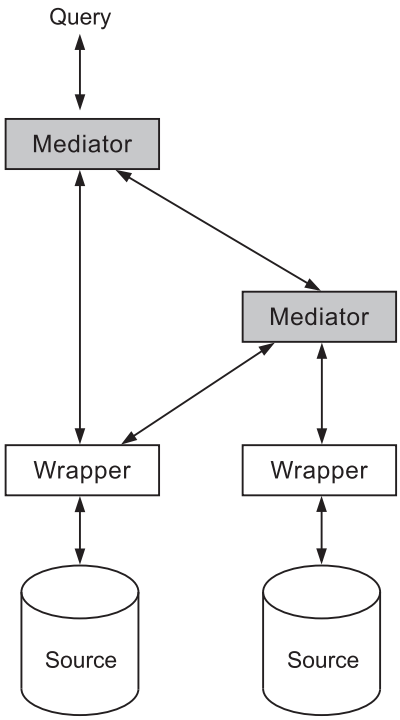

Fig. 1. A mediation integration architecture

A possible way how to describe the mediation system is to use mappings. Mapping is a set of assertions that establish the connection between the element of the global schema and those of the local schema. The composition of mapping is an essential task. It plays a crucial role in query resolving - another important process of a data integration system. Two basic approaches [2], [3] have been used in order to specify the mapping. The *Global As View* (GAV) approach consists in defining the global schema as a set of views over the local schemas, while the *Local As View* (LAV) approach consists in defining the local sources as a set of views made on the global schema.

Because the GAV is based on the idea that the content of each element of the global schema should be characterized in terms of a view over the sources, this mapping tells the system how to retrieve the data. The GAV gives direct information on how query answering may be performed. Some GAV data integration systems do not allow integrity constraints in the global schema. Under these assumptions, query processing can be based on a simple unfolding strategy: every element of the global schema is substituted with the corresponding query over the sources. When global schema allows integrity constraints, the query processing in GAV becomes more complex - integrity constraints here can in principle be used in order to overcome incompleteness of data in the sources.

In GAV query processing can look easy. However, this idea is effective when a set of sources is stable. The addition of a new source and extending the system can be difficult. The new source may have an impact on the definition of various elements of the global schema. So it can force the system designer to redesign the schema, and so to reconsider all the sources.

The LAV approach is based on the idea that the content of each source can be characterized in terms of a view over the global schema. Processing queries in LAV is a difficult task. The only knowledge we have about the data in the global schema is through the views representing the sources, and such views provide only partial information about the data. The mapping specifies the role of each source relation with respect to the global schema. It is not immediate to infer how to use the sources in order to answer queries. The LAV favors the system in the extensibility - addition of sources simply means enriching the mapping with definition of a new view over the global schema, without other changes.

To compensate the insufficiency of the LAV and GAV approaches, also their combinations have been proposed. The *Global Local As View* (GLAV) approach [4] establishes the relationships between the global schema and the sources by making both of LAV and GAV mappings and allows flexible schema definitions independent of the particular details of the sources.

3 Related work

The term ontology [5] has been used in many ways and across different communities. A popular definition of the term ontology in computer science is: an ontology is a formal, explicit specification of a conceptualization. A conceptualization refers to an abstract model of some phenomenon in the world. However, a conceptualization is never universally valid. Ontologies have been set out to overcome the problem of implicit and hidden knowledge by making the conceptualization explicit.

There is a number of approaches to data integration based on ontologies. Ontologies can be employed in various parts of an integration process. At the beginning, ontologies can be used in data sources to describe meaning of the data. They can be used for identification and association of semantically corresponding information concepts. This is crucial in mapping discovery.

There are some projects exploiting from data sources ontologies and staying solving the data integration task with GAV or LAV approaches [6]. However, ontologies take additional tasks in several projects. In some approaches, a global ontology (an ontology of a global view) is built. It can be defined in two basic ways: First, it contains vocabulary shared by local sources. In some of these scenarios, the shared vocabulary is grounding in the particular domain and usually more general then the local ontologies [7]. The other possibility is to build an ontology as a result of local ontologies merging.

Most of data integration projects still stay at definition of mappings as a description of connection between the global view and the local sources, e.g. [8]. Mapping can be done in a broad scale from the simplest one-to-one mapping

rules expressing direct correspondences between terms (synonyms, homonyms, disjoint etc.), trough mapping a concept to a query or a view [9] (like GAV and LAV), to some additional mapping structures (e.g. a reference model in [10]). Some projects use ontology in this part to describe their notion of mapping and then represent mappings as instances in an ontology of mappings.

Approach presented in this paper is similar to approach in [11] - it is also based on merging local ontologies. The difference is that, although a global ontology is a result of merging local ones, they use a mapping table to describe connection between global and local environments. Approach in this paper is based on mapping expression in an ontology that is build by merging global and local ontologies and all relationships as well. Therefore, this approach to data integration is the most similar to projects develop to ontology merging, such as [12].

4 Ontology-based mediation integration

In an ontology-based integration approach described in this paper, a conception of a virtual integration form is adopted. A global source will be also non-materialized and for the establishment of a connection to the data sources some kind of mapping will be applied. However, instead of using mapping rules as assertions for global and local schemas elements, a more complex structure covering all mapping will be employed. This approach exploits the idea that on the Semantic Web [13], [14] every peace of information has got defined its meaning and supposes availability of ontologies as a means for defining the concept of the data. The Semantic Web technique for definition of ontologies is the OWL (Ontology Web Language) language [15], therefore, OWL, as a proposed standard by W3C (WWW Consortium) [16], is used for ontology definition in this approach.

The integration task is transformed to the building of an ontology for the integration system. This ontology from its principle should cover ontologies of all data used in the system and mappings that are in general seen as definitions of relationships between data. Therefore, it can be employed in data integration at schema level.

Ontologies and data schemas are closely related. The main difference is a purpose. An ontology is developed in order to define the meaning of the terms used in some domain, whereas a schema is developed in order to model some particular data. Although it is not necessarily the case that there is some correspondence between a data model and the meaning the terms used, it often does. Especially for schemas representing 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 other cases, an ontology used for that particular data in the source can be enriched by other concepts and relationships to capture also the schema.

Suppose, there are two data sources S_1 and S_2. Each source schema is described by an ontology: an ontology referring to the local source S_1 is O_{S_1}, an

ontology of the source S_2 is O_{S_2}. The global integrated view the integration system should provide has an ontology O_G. The integration system, in Section 2 formalized as a triple $I = (G, L, M)$, has in this case representation

$$I = (O_G, \{O_{S_1}, O_{S_2}\}, O_I),$$

where O_I is *an ontology of the integration system.*

Ontology O_I is used to describe the mapping between elements of the global view and the local sources. Mapping is a crucial part of an integration system and its discovering and expressiveness affects an amount of information we are able to obtain from the local sources via the integration system.

O_I is an ontology of all concepts used in the integration system I. Ontologies O_{S_1} and O_{S_2} are parts of O_I, which can of course be richer. So it follows that for ontologies of local sources is valid:

$$O_{S_1} \subseteq O_I$$
$$O_{S_2} \subseteq O_I$$

While ontologies O_{S_1} and O_{S_2} are given with the sources, O_G and O_I need to be determined. Description of O_G is relatively independent on the sources. O_G contains definition of concepts accessible directly via the global view. It is a matter of a designer who decides what will be accessible via the integration system and in what form. Because O_G describes data of the system, it is a part of O_I, too.

Establishment of O_I is a crucial step. However, it is not an easy task. Even if O_I is used to describe mapping in the integration system, it itself is not a result of mapping source ontologies, but ontology integration. Covering O_{S_1}, O_{S_2}, O_G and their relationships, O_I is the result of task called merging ontologies. The process of ontology merging takes multiple local ontologies as an input and returns a merged ontology as an output. Ontology merging and ontology alignment are widely pursued research topics. Ontology merging is studied e.g. in [17].

Many researchers agree that one of the major difficulty in semantic integration is correspondences discovery. While the formal definitions in an ontology are one of best specifications of the meaning of terms currently available, they cannot capture the full meaning. Therefore, there must often be some human intervention in the process of identifying correspondences. Although machines are unlikely to derive correspondences, it is possible to use them to make suggestions of it and validate human-specified ones.

As in schema integration in other approaches, some conflicts [18] that have to be solved can arise. In general, conflicts can be of various types [19], for example terms conflicts (synonyms, homonyms etc.), schema discrepancies, raw data and metadata conflicts etc. In the ontology world, it is not difficult to deal with different concepts, because there are means how to express relationship between them. In an ontology, each term has a unique reference. Although, there can be two concepts in two ontologies named in the same way, they are uniquely distinguishable, because of the context - the ontology where they are defined.

This is for instance in XML syntax solved by namespaces. Within ontologies it is also possible to state that two terms are equivalent and describe the synonymic relationship, and by this to enable process it in a right way.

Example 1. Suppose, there is a shop that sells notebooks from various producers. For simplification, suppose, there are only two notebook producers: IBM and Sony. Each of them stores data about their products. However, the shop would appropriate to access these sources as one whole, and therefore, there is a need to integrate them.
There are two sources to integrate. Source 1 stores notebooks produced by IBM and its ontology O_1 describes only one class named `IBM_notebooks` with properties `Proc_Speed`, `Memory` etc. Source 2 stores notebooks produced by Sony, its ontology O_2 contain class `Sony_notebooks` with properties `Processor_sp`, `Installed_RAM` etc. Since the integration system should provide notebooks coming from different data sources, global ontology O_G contains class named `notebooks` with properties `Processor_speed`, `RAM`, etc. To obtain ontology of the system, O_1, O_2, O_G, and knowledge about relationships among particular concepts are merged. Ontology O_I is following:

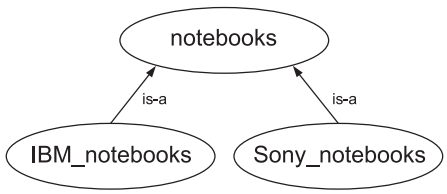

Fig. 2. Ontology O_I

Ontology O_I contains three classes: `notebooks`, `IBM_notebooks`, and `Sony_notebooks`. Notebooks from IBM and notebooks from Sony are both notebooks, so there is hierarchical class - subclass relationship between `notebooks` and `IBM_notebooks` and between `notebooks` and `Sony_notebooks`. `IBM_notebooks` and `Sony_notebooks` cannot be merged into one class, because it refers to different notebooks. With the knowledge of class properties semantics, there can be seen property - subproperty relationship between a global property and a relevant local property, for example `Processor_speed` and `Proc_Speed`. Moreover, if there were the same integrity constraints on each property from the pair, the properties can be merged and connected as equivalent. □

With a data integration system, a user poses his query on the global view in terms of the global view. In order to execute the query over the sources, where data are stored, query processing 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 logically implies that it is an answer to the query.

With mapping expressed in an ontology, considering only hierarchical is-a relationship, it is possible for query rewriting to adopt a rule well known in object-oriented world: a child can substitute his parent. If we are looking for all instances of class C that have property $P = x$, the query is

$$q := C(P = x).$$

Using ontology O_I, is-a hierarchy relationships give a means how to rewrite the query with respect to a specific local source. If C is not a concept of the local source schema, class C in the query is replaced with its nearest subclass C' in the is-a hierarchy. This is recursively repeated until a concept is founded in the specific local source schema, or there are no more subclasses - there is no answer. The same rule as for classes can be adopted also for properties, and the relationship property-subproperty can be employed.

In query answering approach to query processing, the is-a hierarchy is also essential. It expresses that an instance of a node is an instance of all nodes within the path from the root node to this node. Based on this rule, it can be determined if information from a local source can be an answer to the global query.

Example 2. Continuing the simple example of notebooks integration, this example shows query processing. The global view provides notebooks. The query: give all available notebooks with processor speed 1.6 GHz, i.e.

$$q := notebooks(Processor\_speed =' 1.6'),$$

is processed as follows: `notebooks` is not in the concept of any local source, the query is rewritten. `notebooks` class has two child nodes `IBM_notebooks` related to the source 1 and `Sony_notebooks` related to the source 2. The reformulated query has two forms:

$$q_1' := IBM\_notebooks(Processor\_speed =' 1.6')$$

and

$$q_2' := Sony\_notebooks(Processor\_speed =' 1.6').$$

Because the property `Processor_speed` is not in the concept of the source 1, the query q_1 is further rewritten using property-subproperty to

$$q_1'' := IBM\_notebooks(Proc\_Speed =' 1.6').$$

The query q_1'' is executed over the source 1. Analogously, the query q_2' is rewritten and executed over the source 2. □

However, ontology is more powerful than to express only is-a relationships. Ontology O_I can contain various kinds of relationships. Concepts can be less related, or their relationship can depend on other circumstances. For example, in order to obtain as much information from the sources for the particular query as possible, it could be also appropriate to use some inference mechanism in query processing. Therefore, for future work it is planned to work further on other concepts relationships and their use in query processing.

Compared with two basic approaches of mapping specification in a mediation data integration, an ontology-based approach is similar to LAV integration in a way, that the global schema is specified independently from the sources. Another similarity can be found in extending the system. When a new source is added, the ontology of the integration system O_I is enriched with a new source ontology and further possible relationships to previous version of O_I. A difference between LAV and GAV and the ontology-based integration system is in the case of a change in the layer of local or global source schemas. In case of using ontologies, the ontology of integration system is enriched with the new state. It is not needed to change any earlier part of the ontology, or even to remove some part. No other change is needed.

5 Conclusion

Data integration is a task of combining data from different data sources and enabling a user to process them as one whole. There are two classical ways of designing an integration system providing a global virtual view over the sources: GAV and LAV approaches. Both are based on definition of connection between the global view and the local sources via mappings as views. With a Semantic Web idea, there are also other possibilities that can be used. A number of approaches exploit from the available ontologies describing data in integrated sources. An integration system described in this paper is also based on ontologies of the sources. Moreover, it uses an ontology also for mapping description. This brings not only the possibility to capture various kinds of concept correspondences, but also a possibility to reuse it in other tasks or situations. However, a variability of captured correspondences rise a need to use an inference mechanism in query processing part of integration.

Acknowledgements

This 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 by the Institutional Research Plan AV0Z10300504 “Computer Science for the Information Society: Models, Algorithms, Applications”.

References

1. J. D. Ullman, "Information integration using logical views", *Theoretical Computer Science* 239 (2000), pp. 189-210.
2. 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.
3. 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.
4. M. Friedman, A. Levy, and T. Millstein, "Navigational plans for data integration", In Proceedings of the *16th Nat. Conf. On Artificial Intelligence (AAAI'99)*, AAAI Press, pp. 67-73, 1999.
5. Y. Ding, D. Fensel, M. Klein, and B. Omelayenko, "The semantic web: yet another hip?", *Data & Knowledge Engineering* 41 (2002), pp. 205-227.
6. B. Amann, C. Beeri, I. Fundulaki, and M. Scholl, "Ontology-Based Integration of XML Web Resources", In Proceedings of the *1st Int. Semantic Web Conference (ISWC 2002)*, pp. 117-131, 2002.
7. N. F. Noy, "Semantic Integration: A Survey Of Ontology-Based Approaches," In *ACM SIGMOD Record, Special Section on Semantic Integration*, vol.33, 4, pp. 65-70, 2004.
8. Z. Cui, D. Jones, and P. O'Brien, "Issues in Ontology-based Information Integration", In Proceedings of the *IJCAI Workshop: Ontologies and information sharing*, Seattle, USA, 2001.
9. 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.
10. 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.
11. I. F. Cruz, H. Xiao, and F. Hsu, "An Ontology-based Framework for XML Semantic Integration", In Proceedings of the *8th Int. Database Engineering and Application Symposium (IDEAS'04)*, Coimbra, Portugal, pp. 217-226, 2004.
12. N. F. Noy and M. A. Musen, "The PROMPT suite: Interactive tools for ontologzy merging and mapping.", *International Journal of Human-Computer Studies* vol. 56, 6, pp. 983-1024, 2003.
13. T. Berners-Lee, J. Hendler and O. Lassila, "The Semantic Web", *Scientific American*, vol. 284, 5, pp. 35-43, 2001.
14. M.-R. Koivunen and E. Miller, "W3C Semantic Web Activity", in the proceedings of the *Semantic Web Kick/off Seminar*, Finland, 2001.
15. Web Ontology Language (OWL), `http://www.w3.org/2004/OWL`.
16. W3C (WWW Consortium), `http://www.w3.org`.
17. K. Kotis, G. A. Vouros, and K. Stergiou, "Towards automatic merging of domain ontologies: The HCONE-merge approach", *Web Semantics: Science, Services and Agents on the World Wide Web* 4 (2006), pp. 60-79.
18. S. Bergamaschi, S. Castano, M. Vincini, and D. Beneventano, "Semantic integration of heterogeneous information sources", *Data & Knowledge Engineering* 36 (2001), pp. 189-210.
19. C.-Y. Lee and V.-W. Soo, "The conflict detection and resolution in knowledge merging for image annotation", *Information Processing and Management* 42 (2006), pp. 1030-1055.