Integrating Federated Information Sources through the use of Ontologies

Agustina Buccella, Alejandra Cechich
Departamento de Ciencias de la Computación, Universidad Nacional del Comahue,
Buenos Aires 1400, Neuquén, Argentina
Email: abuccel,acechich@uncoma.edu.ar

Miguel R. Penabad, Francisco J. Rodriguez, and Nieves R. Brisaboa
Departamento de Computación, Universidade Da Coruña,
Campus de Elviña s/n, 15071 A Coruña, Spain España
Email: penabad@udc.es, franjrm@uvigo.es, brisaboa@udc.es

Abstract. Ontologies serve as a mean to establish a conceptually concise basis for communicating knowledge for many purposes. Within the information integration area, ontologies are useful to solve heterogeneity problems. They provide a set of knowledge terms, including the vocabulary, the semantic interconnections and some simple rules of inference and logic. In this paper we present a helpful method to integrate a new data source into a federated system taking advance of the use of ontologies.

Keywords: Federated Systems, Ontology, Data Integration.

1. Introduction

Several aspects must be taken into account when working with Federated Systems because the main characteristics of these systems make difficult the integration tasks. For example, the autonomy of the information sources, their geographical distribution and the heterogeneity among them, are some of the main problems we must face to perform the integration [12]. Particularly, the semantic heterogeneity has been one of the most researched aspects in the last years. Works like [9,14] are aimed to fill the semantic gap among the information sources. To do so, several works use the semantic information the ontologies provide.

We are currently working on some of these aspects in order to facilitate the integration task. In recent works [4,5,6,7] we have proposed an architecture and a method to solve heterogeneity problems [18] among a fixed set of data sources. To do so, in this work we assume the integrated system has been created by following a series of rules defined in [5].

Figure 1 shows a part of the architecture of our integrated system based on a hybrid ontology approach [19]. As we can see, the architecture is composed of a *global ontology* or *shared vocabulary* containing the generic concepts that will be used to query the system. Users use the vocabulary to query and get answers to the system. Another component is the *Ontology Mapping (OM)* component which deals with the information flow between the source ontologies and the shared vocabulary. This component contains a set of mappings relating concepts in the *sources ontologies* with concepts in the shared vocabulary. Once the user chooses the concepts from the shared vocabulary and makes the query, the system uses the OM component to know which concepts are related with. Thereby, through the sources ontologies, the system gets access to the information sources to produce the data. There is only one source ontology for each information source.

This paper is organized as follows. Section 2 discusses our method to add a new information source in the federated system. By a series of stages we will explain the needed tasks in order to achieve a consistent integration. The method modifies and adds information to the three

components of the architecture in order to store the information the new sources provide. Conclusions are shown in Section 3.

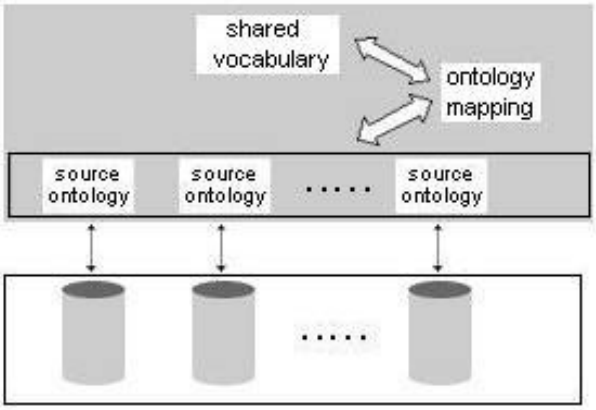

Figure 1. A part of our integration system

2. A Data Integration Process

The construction of a federated system is not a straightforward task, so we propose a guide in order to do this activity more consistent and correct. Nowadays, many enterprises have their data stored on relational databases and therefore we assume the data sources are relational database systems. When a new database enters to the integrated system, a series of tasks must be followed. There are two main stages involved in this process – *building the source ontology* and *building the mappings between the source ontology and the shared vocabulary*. In order to achieve these two stages a set of steps should be performed. We briefly explain each of them in order to clarify the process.

The first stage, *building the source ontology*, contains two main steps: *generating the initial ontology* and *adding semantics*.

The first step stage takes as input a relational model [8] and automatically generates an initial ontology. In order to build the initial ontology, we use the information provided by the Data Definition Language code (SQL/DDL) because it is always provided by the database. It is very common the documentation about the domain represented by the databases is not available. But the relational databases have a mechanism to extract the relational tables stored physically. Therefore, we assume the DDL-code is the available information about the domain which can be used to create the initial ontology. By a series of rules we transform this code in an ontology. This ontology is represented by the Web Ontology Language (OWL) [17,1]. We have chosen OWL due to its widespread use in the Semantic Web [3]. Besides, OWL allows us to formalize a domain by defining classes and properties of those classes, to define individuals asserting properties about them, and to reason about these classes and individuals to the degree permitted by the formal semantics of the OWL language. OWL can be (partially) mapped onto a description logic [2] making possible the use of existing reasoning tools such as FACT [13] and RACER [11].

The second step, *adding semantics*, allows the expert (for example, using an ontology editor as Protégé [10] with the OWL plug-in) to add restrictions, classes and/or properties to the initial ontology. Knowing the domain of the information source and understanding the structures, the user is able to provide more semantics to the ontology. This step is performed by assisting expert users.

The second stage, *building the mappings between the source ontology and the shared vocabulary*, contains three main steps: *searching for similarities*, *adding mapping into the OM* and *adding the new information to the shared vocabulary.*

The first step, *search for similarities*, implies searching for similarities between the concepts. In previous works [4,5] we have shown the use of two similarity functions proposed in [15,16] in order to find similarities among the classes and properties. The system will use the similarity function in order to propose to the user the possible comparison among classes and properties. That is, the system guides the user to perform this step. So, the following process is performed: (1) the expert user chooses one class or property of the source ontology and one class or property of the shared vocabulary in order to indicate the similarity between them to the system; (2) then, taking into account this first similarity, using the similarity functions and following the related classes, the system proposes other similarities to the user. This process can be repeated several times until all the classes and properties are compared. So, the expert user will make the last decision.

The following two steps, *adding mappings into the OM* and *adding the new information to the shared vocabulary* are automatic steps, that is, we will create a system that implements these steps without user intervention. The former step is achieved by using the similarities found in the last stage. The OWL ontology mapping constructors will be used to store the mappings in the OM component. The latter step, *adding the new information into the shared vocabulary*, consists of adding the information the shared vocabulary does not contain but it is provided by the source ontology. Thus, the shared vocabulary will make available all the information the sources ontologies offer.

3. Conclusions

We have proposed a method to integrate information sources. Our method takes the advantages of the ontologies in order to solve heterogeneity problems. Both stages of our method have a different level of complexity and must be analyzed separately. We are working on the construction of several automated and not automated tools in order to accelerate this process. Each step of our method is accompanied by one of these tools. At the moment we are developing tools for the first step, however several works related with the second stage are already under development.

4. References

1. Antoniou, G., Harmelen F. Web Ontology Language: OWL. Handbook on Ontologies in Information Systems. Staab & Studer Editors. Springer-Verlag, 2003.
2. Baader, F., Calvanese, D., McGuiness, D., Nardi, D. and Patel-Schneider, P. editors. The Description Logic Handbook - Theory, Implementation and Applications. Cambridge University Press, ISBN 0-521-78176-0, 2003.
3. Berners-Lee, T. XML 2000 – Semantic Web talk, 2000 http://www.w3.org/2000/Talks/1206-xml2k-tbl/slide10-0.html.
4. Buccella A., Cechich A. and Brisaboa N.R. An Ontology Approach to Data Integration. Journal of Computer Science and Technology. Vol.3(2). Available at http://journal.info.unlp.edu.ar/default.html, 2003, (pp. 62-68).
5. Buccella A., Cechich A. and Brisaboa N.R. An Ontological Approach to Federated Data Integration. 9° Congreso Argentino en Ciencias de la Computación, CACIC'2003, La Plata, October 6-10, 2003, (pp. 905-916)..
6. Buccella A., Cechich A. and Brisaboa N.R. A Context-Based Ontology Approach to Solve Explanation Mismatches. Jornadas Chilenas de Computación. JCC 2003. Chillán, Chile, November 3-9, 2003.

7. Buccella A., Cechich A. and Brisaboa N.R. An Ontology-based Environment to Data Integration. VII Workshop Iberoamericano de Ingeniería de Requisitos y Desarrollo de Ambientes de Software. To appear in IDEAS 2004. 3-7 de Mayo, 2004.
8. Codd, E. A Relational Model of Data for Large Shared Data Banks. *Communications of the ACM*, Vol.13(6), 1970, (pp. 377-387).
9. Euzenat, J., Valtchev, P. An integrative proximity measure for ontology alignment. CEUR Workshop Proceedings. Sanibel Island, Florida, October 20, 2003.
10. Gennari, J., Musen, M. A., Fergerson, R. W., Grosso, W. E., Crubézy, M., Eriksson, H., Noy, N. F., Tu, S. W. The Evolution of Protégé: An Environment for Knowledge-Based Systems Development. Technical Report, SMI-2002-0943, 2002.
11. Haarslev, V. and Moller, R. RACE system description. In P. Lambrix, A. Borgida, M. Lenzerini, R. Moller, and P. Patel-Schneider, editors, Proceedings of the International Workshop on Description Logics, number 22 in CEUR-WS, Linkoeping, Sweden, July 30-August 1 1999, (pp. 140-141).
12. Hasselbring, W. Information System Integration. Communications of the ACM. June 2000.
13. Horrocks, I. The FaCT system. In H. de Swart, editor, Automated Reasoning with Analytic Tableaux and Related Methods: International Conference Tableaux'98, number 1397 in Lecture Notes in Artificial Intelligence, pages 307--312. SpringerVerlag, Berlin, May 1998.
14. Maedche, A. and Staab, S. Measuring Similarity between Ontologies. In: Proc. Of the European Conference on Knowledge Acquisition and Management - EKAW-2002. Madrid, Spain, October 1-4, 2002. LNCS/LNAI 2473, Springer, 2002, (pp. 251-263)..
15. Rodriguez, A., Egenhofer, M. Determining Semantic Similarity among Entity Classes from Different Ontologies. IEEE Transactions on Knowledge and Data Engineering, vol. 15, no. 2, March/April 2003.
16. Rodriguez, A., Egenhofer, M. Putting Similarity Assessments into Context: Matching Functions with the User's Intended Operations. Context 99, Lecture Notes in Computer Science, Springer-Verlag, September 1999.
17. Smith, M.K.,Welty, C., McGuinness, D.L. OWL Web Ontology Language Guide. W3C, http://www.w3.org/TR/2004/REC-owl-guide-20040210/. 10 February 2004.
18. Visser, P., Jones, D., Bench-Capon, T., Shave, M. An Analysis of Ontology Mismatches; Heterogeneity versus Interoperability. AAAI 1997 Spring Symposium on Ontological Engineering.
19. Wache, H., Vögele, T., Visser, U., Stuckenschmidt, H.. Schuster, G., Neumann, H. and Hübner, S. "Ontology-based Integration of Information - A Survey of Existing Approaches," In: Proceedings of IJCAI-01 Workshop: Ontologies and Information Sharing, Seattle, WA, Vol. (Pages 108-117). 2001