J Data Semant (2013) 2:57–59
DOI 10.1007/s13740-013-0020-6

EDITORIAL

Special Issue on: Evolution and Versioning in Semantic Data Integration Systems

Ladjel Bellatreche · Robert Wrembel

Published online: 17 May 2013

1 Introduction

Data integration systems aim at integrating data from multiple heterogeneous, distributed, autonomous, and evolving data sources (DSs) to provide a uniform access interface to end users. Typically, integration systems are based on the three following architectures: materialized (where data sources are duplicated in a repository), virtual (where data are kept in their sources), and hybrid (which combines the two former ones). A good example of the materialized architecture is a data warehouse (DW), which is dedicated for business applications. A DW includes different components: an DSs layer, and extraction-transformation-loading (ETL) layer, a DW layer, and an on-line analytical processing (OLAP) layer. In the virtual architecture, a special component, called a mediator, provides an integrated view (a global schema) on the source schemas. User queries are expressed in terms of the global schema. A mediator provides a virtual database, translates user queries into specific queries on DSs, synthesizes the results of these queries, and returns answers to a user.

One of the main difficulties of building data integration systems is the heterogeneity of data sources. The semantics of data sources is usually implicit or unknown. Most DSs participating in the integration process were designed to satisfy day-to-day applications and not to be integrated in the future. Often, the small amount of semantic contained in their conceptual models is lost, since only their logical models are implemented and used by applications. The presence of a conceptual model may allow designers to express the application requirements and domain knowledge in an intelligible form for a user. Thus, its absence or any other semantic representation in final databases makes their interpretation and understanding complicated, even for designers who have good knowledge of the application domain. The heterogeneity of data sources impacts both the structure and the semantic. To deal with semantic problems and ensure an automatic data integration, a large number of research studies propose the use of ontologies to describe the semantic of various sources in data warehouse and mediator architectures. Ontologies showed their efficiency in materialized and virtual data integration systems. Recently, the database community proposed solutions for building semantic DWs from sources referencing domain ontologies.

Methods used for designing semantic integration systems, research developments, and most of the commercially available technologies tacitly assumed that a semantic integration system is static. In practice, however, this assumption turned out to be false. A semantic integration system requires changes among others as the result of: (1) the evolution of DSs, (2) changes of the real world represented in an integration system, (3) the evolution of domain ontologies referencing sources and a local ontology of the semantic integration system, (4) new user requirements, and (5) creating simulation scenarios (what-if analysis). As reported in the literature, structures of data sources change frequently. For example, during the last 4 years, the schema of Wikipedia changed every 9–10 days, on the average. From our experience, schemas of EDSs may change even more frequently. For example, telecommunication data sources changed their schemas every 7–13 days, on the average. Banking data sources are more stable but they changed their schemas every

L. Bellatreche
Ecole Nationale Supérieure de Mécanique et d'Aérotechnique, Poitiers, France
e-mail: bellatre@ensma.fr

R. Wrembel (✉)
Poznan University of Technology, Poznan, Poland
e-mail: Robert.Wrembel@cs.put.poznan.pl

2–4 weeks, on the average. Changes in the structures of DSs impact all the layers of the semantic integration system. Since such changes are frequent, developing a technology for handling them automatically or semi-automatically in a semantic integration system is of high practical importance.

Existing approaches to handling the evolution of an integration system in general and data warehousing in particular can be categorized as: (1) ETL evolution, (2) data warehouse evolution, and (3) the evolution of optimization structures. ETL evolution has not received many attention from the research community so far. The most advanced approach, i.e., Hecateus, solves the problem only partially. Still open issues in this area concern: modeling ETL workflows, designing taxonomy and rules for ETL evolution, deploying these rules, and plugging in the evolution techniques into existing ETL engines. The most eligible approaches to handling a semantic integration system evolution are based on schema evolution techniques, or on temporal extensions, or on versioning. Schema evolution techniques are able to represent only the current integration schema and data, i.e., historical integration schema states are lost. Temporal extensions are able to handle multiple historical states of data but they manage one invariant global schema. In versioning techniques, evolving data and schemas are managed partially by means of schema versions and partially by data versions. The versioning techniques, although the most promising, still need development. They posses limited capabilities of querying integration versions, and few index structures for multiversion data have been developed. Moreover, there is a need to integrate temporal extensions (for managing data evolution) and versioning techniques into one consistent framework.

The selection of optimization structures during the physical design (materialized views, caching techniques, partitioning, indexing, parallelization, etc.) is usually done in a static way. The evolution of different components of semantic integration systems has a strong impact on the final optimization structures. The dependencies between these components and their evolution impact on an integration system's performance have to be further explored.

The aim of this special issue of the *Journal on Data Semantics* is twofold: First, to present new and challenging issues on evolution management and versioning in semantic integration systems. Second, to present the current research and technological developments in this field.

2 Content

The issue is composed of five papers that, in general, address the following research issues: business process evaluation based on ontologies, ontology modeling, as well as constructing and managing the evolution of ontologies. The first three papers are the extensions of the best data semantic papers selected from the Advances in Databases and Information Systems—ADBIS conference (held on September 18–20, 2012, Poznań, Poland).

The paper entitled *A Framework for Alignment of Data and Processes Architectures Applied in a Government Institution*, by C. Castellanos and D. Correal, presents a framework for a verification of business processes in a given organization with the underlying IT architecture, i.e., checking whether the processes are supported by the IT architecture. The verification is automatically executed by means of ontology matching that describes both the processes and the IT architecture. To this end, the authors extended the metamodel of an enterprise architecture with associations between entities and business processes. Then, they developed a procedure for the alignment of business process and data elements based on ontology matching. The procedure lists all the alignments and misalignments. They further can be queried by means of a query language, called Kalcas.

The paper entitled *Improving Business Process Model Quality Using Domain Ontologies*, by S. Si-Said Cherfi, S. Ayad, and I. Comyn-Wattiau, discusses the problem of improving a quality of business process models. To this end, the authors proposed to support modeling with ontologies describing a modeled domain. The proposed approach is supported by meta-models that describe both a domain ontology and a business process model, the alignment between the metamodel elements, and a set of the Object Constraint Language mapping rules that map the ontology entities to their corresponding business process entities.

The paper entitled *Grounding Ontologies with Social Processes and Natural Language*, by C. Debruyne, T.K. Tran, and R. Meersman, presents the solution for managing hybrid ontologies, i.e., ontologies whose concepts are described formally and informally. The formal descriptions are based on the so-called fact-oriented ontology framework where the knowledge building blocks are binary fact types, also grounded in natural language. The informal descriptions are supported by a glossary. Hybrid ontologies are constructed and managed in a framework, called GOSPL. The paper also describes how the evolution of glosses impacts the formal definition of the ontology and how a single application commits to an ontology. All these issues are illustrated with a prototype software.

The paper entitled *Ontology Change Management and Identification of Change Patterns*, by M. Javed, Y.M. Abgaz, and C. Pahl, presents an approach to handling the evolution of ontologies. It is based on the four following features: (1) change operationalization, (2) change representation, (3) change semantic capturing, and (4) change pattern discovery. The approach is built on a layered change log model where ontology engineers typically deal with generic changes in the first two upper levels, whereas other users (e.g., domain experts, content managers) work at the lower

level. The change log was formalized using a graph-based approach in order to support a discovery of ontology change patterns. The pattern identification algorithms were shown and their performance was evaluated.

The paper entitled *DYNAMO-MAS : a multi-agent system for ontologies evolution from texts*, by Z. Sellami, V. Camps, and N. Aussenac-Gilles, presents the experience of the authors in constructing and evolving ontologies from texts. The proposed framework and prototype software, called DYNAMO-MAS, allows to propose to a user the ontological concepts, validates them, and learns the process of constructing an ontology. Its implementation is based on the so-called adaptive software agents that represent an evolving ontology and the linguistic data extracted from the texts.

Acknowledgments The guest editors would like to acknowledge the help of all involved in the review process of this special issue of the *Journal on Data Semantics*. The reviewers provided comprehensive, critical, and constructive comments. Without their support, the project could not have been completed. The alphabetically ordered list of reviewers includes: Alberto Abello, Universitat Politècnica de Catalun-ya, Spain; Yamine Aït Ameur, Ecole Nationale Supérieure de Mécanique et d'Aérotechnique, France; Sonia Bergamaschi, Università degli Studi di Modena e Reggio Emilia, Italy; Omar Boussaid, Université Lyon 2, France; Dickson Chiu, Chinese University of Hong Kong, China; Alfredo Cuzzocrea, ICAR-CNR and University of Calabria, Italy; Marcin Gorawski, Silesian University of Technology, Poland; Jorge Gracia, Universidad Politécnica de Madrid, Spain; Stéphane Jean, Ecole Nationale Supérieure de Méca-nique et d'Aérotechnique, France; Haridimos Kondylakis, Foundation of Research and Technology-Hellas (FORTH), Greece; Jens Lechtenbörger, Westfälishe Wilhelms-Universi-tät Münster, Germany; Brahim Medjahed, University of Michigan, Dearborn, USA; Chantal Reynaud, LRI-Université Paris-Sud, France; Dimitrios Skoutas, University of the Aegean, Greece; David Taniar, Monash University, Australia; Leandro Krug Wives, Universidade Federal do Rio Grande do Sul, Brazil; Oscar Romero, Universitat Politècnica de Catalunya, Spain. The editors would also like to thank Professor Esteban Zimányi for accepting our proposal of this special issue. Poitiers, France and Poznań, Poland, January 2013
Ladjel Belletreche and Robert Wrembel