Semantic Integration of Relational Data Sources With Topic Maps

Thomas Neidhart, Rani Pinchuk, and Bernard Valentin

Space Applications Services, Leuvensesteenweg 325, 1930 Zaventem, Belgium
{thomas.neidhart,rani.pinchuk,bernard.valentin}@spaceapplications.com

Abstract. Data integration of heterogeneous data sources plays a major role in the development of modern knowledge management systems. Additional enrichment of this data with the use of ontologies opens up completely new possibilities in leveraging the use of semantic technologies, and combining information from existing information systems. This paper presents the architecture and prototype implementation of a semantic integration layer for transparent access to relational data sources through the use of Topic Maps.

1 Introduction

Nowadays, when the rate in which data is gathered is higher than the rate it can be processed by humans, one of the main obstacles in processing the data available is the fact that the data is usually spread across heterogeneous data sources with different database designs and access methods.

Because the data available is not self-explanatory, when accessing a data store, its design must be well understood. This becomes a much bigger problem when several databases have to be accessed [9].

With the emergence of Semantic Web technologies, the demand for building semantic-aware knowledge management systems is becoming more and more a necessity. Technologies like RDF, OWL or Topic Maps have been developed to represent knowledge in a way that makes it human- and/or machine-understandable. Such technologies can be used in order to annotate data with metadata information and thus enabling the integration on a semantic level.

However, the details of wrapping available data with semantics remains open. Because the amount of data is huge, and many times, also due to policy related reasons, copying the data is out of the question. A semantic view over the available data has to be created, providing a semantic integration layer.

Multiple ontology-based approaches for data integration [12] have been researched in the past. This paper presents a prototype of a semantic integration layer built on top of an open source Topic Maps engine [11].

The remainder of the paper is structured as follows. First we present the proposed mapping between relational data structures and Topic Maps in section 2, and outline our Semantic Integration Architecture in section 3. Section 4 discusses selected implementation details with references to the project that

Maicher, L.; Garshol, L. M. (Eds.): Linked Topic Maps. Fifth International Conference on Topic Maps Research and Applications, TMRA 2009 Leipzig, Germany, November 12–13, 2009 Revised Selected Papers. Leipziger Beiträge zur Informatik. ISBN 978-3-941608-06-1

motivates this approach. Some related work is listed in section 5, and the paper is concluded with final comments in section 6.

2 Mapping of Relational Data Structures to Topic Maps

Based on the notation defined in the Topic Maps – Data Model [4], this section presents the proposed mapping between arbitrary relational data structures to Topic Maps constructs (please note that basically any possible mapping between relational data structures and Topic Maps Constructs is valid, the presented mapping is limited to the most common combinations to illustrate the concept in a clean and structured fashion).

Relational database structures are typically composed of a set of tables, where each table consists of certain elements:

- Primary Key: uniquely identifies a data element in a database table.
- Column: contains information associated of a data element.
- Foreign Key: relates data elements with each other within a database.

From an abstract viewpoint, Topic Maps are quite similar to general relational data structures, but they have more advanced possibilities to identify a data element, and to add semantic meaning to it.

- Topic: represents a data element in a Topic Map.
- Item Identifier: uniquely identifies a Construct[1] within a Topic Map.
- Subject Id./Locator: uniquely identifies a Topic within a Topic Map.
- Name: associates a common name to a Topic.
- Variant: associates a different form of a name to a Topic.
- Occurrence: contains information about a data element (Topic).
- Association: relationship between two or multiple Topics.
- Role: part of a single Topic within one Association.

A possible mapping (see Fig. 1) between relational data structures and Topic Maps would include the following:

- Primary Key Item Identifier/Subject Locator/Subject Identifier
- Column Name, Occurrence or Variant
- Foreign Key Role

With such a mapping, the information stored in a relational data structure can be mapped into Topic Maps notation. Additional semantic information (as typically found in Topic Maps) can be defined in advance (see section 4.1) to include such data elements into a type hierarchy system.

[1] A Topic Maps Construct [4] is the collective for the concepts Topic, Association, Name, Variant, Role and Occurrence.

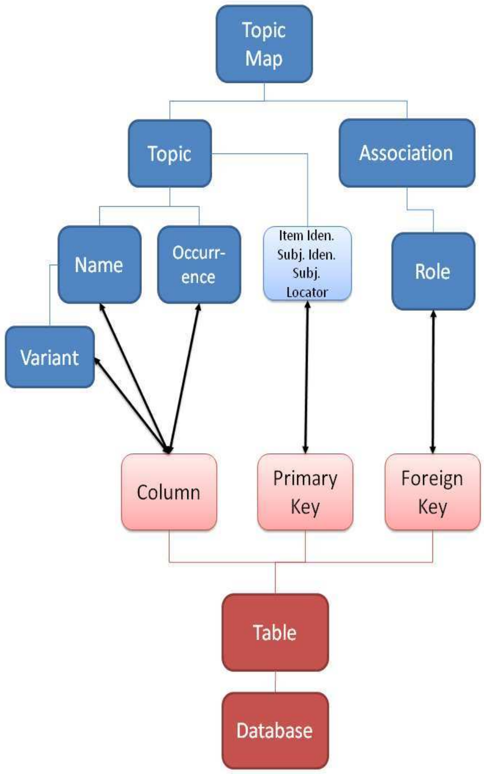

Fig. 1. A mapping of relational data structures to Topic Maps Constructs

3 Architectural Overview

The architecture (see Fig. 2) for the proposed semantic integration layer is built around a freely available Topic Maps engine [11] which has been developed by Space Applications Services.

A Topic Maps engine provides an interface for accessing, modifying and storing Topic Maps, and in our case, also provides access to data that resides in heterogeneous data stores. Whereas the Topic Maps engine provides a unified, structured view on the data from different sources, all data remains at their origin.

Clients can access the unified topic map either by using the TMQL query interface [6] or programmatically through a TMAPI [3] compatible interface. The ontology definition which is loaded by the Topic Maps engine contains only the Topic Maps ontology, that is, the types of the topics, occurrences and associations which practically constitute the structure of the topic map.

Access to external data sources is available through so called "c**onnectors**", which can be attached to the Topic Maps engine. Each connector contains a

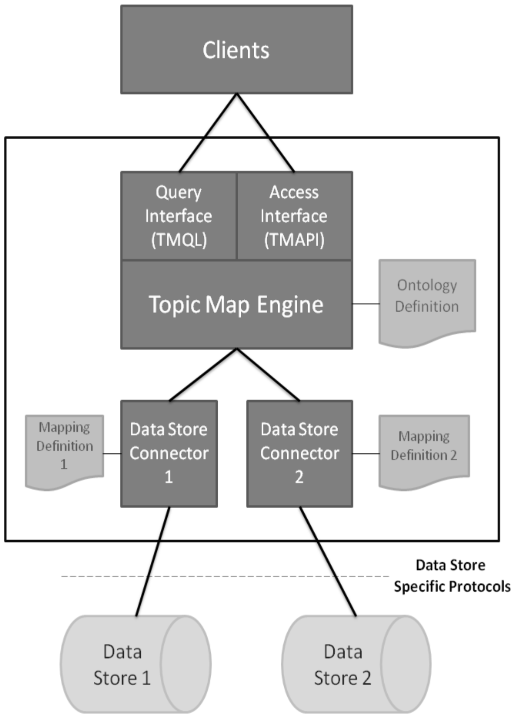

Fig. 2. Semantic Layer Architecture

"**mapping definition**", describing which type of data source the connector is attached to, and how the contents of the data source are mapped to appropriate Topic Maps constructs (e.g. topics, names, occurrences, and associations). With the use of such a connector, one can create therefore a "live" view of the attached data source, which is accessible through a Topic Maps interface, annotating its data with semantic information.

When accessing a topic map with multiple **connectors** attached, the Topic Maps engine will access all suitable connectors in order to find the relevant information, the client is looking for. The described architecture is therefore capable to seamlessly merge data from multiple attached connectors together with ordinary Topic Maps.

4 Implementation Aspects

In this section, the implementation details of the proposed semantic layer are presented. The mapping definition is detailed as well as the supported data sources so far. The presented work is based on a prototype of the proposed semantic integration layer and is limited in certain aspects (see section 6) but nevertheless illustrates the basic concept and the use in a real-world scenario.

4.1 Mapping Definition

The mapping definition file describes how information stored in a relational data source should be mapped into appropriate Topic Maps constructs.

The syntax for defining such a mapping has been derived from the Topic Map Compact Syntax Standard [5], as this provides an intuitive and declarative way to describe topic maps. Topics can be described with just a few syntactical elements, while identifiers follow a special scheme in order to conform to the CTM grammar:

- Names: a dash followed by a quoted string
- Occurrences: name/value pairs
- Item Identifiers: followed by the id
- Types: isa followed by the type
- Associations:
  `association-type (role1:player1, ..., roleN:playerN)`

With these simple constructs, it is possible to create a meaningful mapping definition, where each topic map construct is mapped to an information element in the attached data source.

An example mapping file for a relational database is provided in the following code snippet:

```
%ctm 1.0
%prefix database <territories.db>
%prefix tablename <subbasin>

topic - "${subbasin.NAME}";
  ^ ${subbasin.TMID};
  isa sub-basin;
  size: "${subbasin.AREA_M2}";
  latitude: "${subbasin.LAT}" .
  is-located-in(location: topic, host: "${subbasin.BASIN_TMID}")
```

This defines a mapping for *topics* of type **sub-basin** which are stored in a SQLite database. The *name* and *item identifier* for the *topics* are located in the columns **NAME** and **TMID**. Additionally there are two occurrences defined (size and latitude), which are mapped to the appropriate table columns in the database. For each topic, an *association* will be defined, that gives information in which *basin* a specific *sub-basin* **is-located-in**. The association will be dynamically created, based on the information in the *BASIN_ TMID* column of the specific topic.

A unique feature of Topic Maps is the so called Item Identifier. An Item Identifier is essentially a well-known identifier for a Topic Maps Construct. When

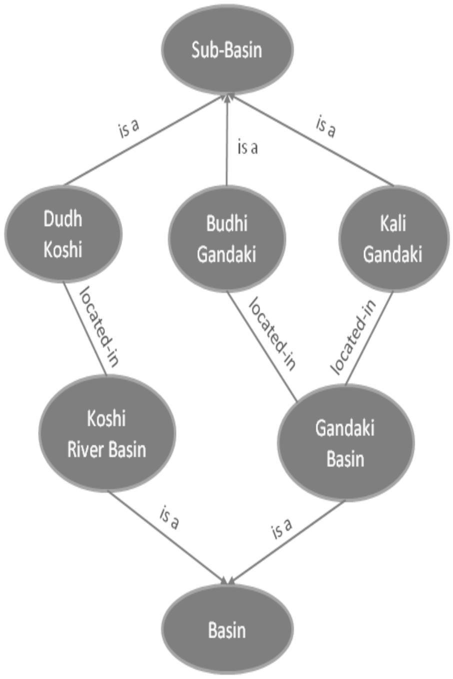

Fig. 3. Unified Topic Map Example

connecting to external data stores, an item identifier will be used to uniquely identify an information element within the data store (e.g. the primary key in a database table).

An excerpt of a unified topic map, created with the sample mapping definition described before can be seen in Fig. 3. There one can see three sub-basins which have an association to the basin they are actually located-in. The information about the sub-basins comes from the data store, is transformed into Topic Maps constructs and enriched with semantic data (type hierarchy, associations).

With such a mapping definition, it is very easy to wrap a custom knowledge structure around existing data without modifying legacy data stores.

4.2 Data Sources

Vast amounts of data are stored in countless data sources, including text documents, CSV files, XML data, databases, web pages and information systems (e.g. Lotus Notes).

For the prototype implementation we focused on supporting simple relational data sources (e.g. **SQLite** databases). Supporting different types of data sources will be possible, but needs to take caching into account, as the lack of efficient query facilities could have a major impact on the overall performance of such a system.

The current implementation for accessing information stored in database tables follows the Proxy Pattern. Whenever a method of the Topic Maps engine is called, that would access data stored in a connector, the appropriate Proxy class generates SQL statements "on-the-fly", executes them and returns the results as Topic Maps constructs.

5 Related Work

Several research groups have recently studied the use of Topic Maps based approaches to integrate distributed knowledge models.

The proposed approaches range from using Topic Maps technology as a middleware framework for accessing knowledge models [1], implementing a protocol to remotely access Topic Maps [2] to introducing the concept of a Topic Grid to leverage the use of distributed knowledge models for the development of semantic applications [7]. A conceptual predecessor of the presented work is the DB2TM tool of the Ontopia Knowledge Suite [10], allowing to fully migrate relational databases into Topic Maps.

6 Conclusions

In this paper we have presented the architecture and a prototype implementation for a heterogeneous knowledge management system infrastructure, which allows a client to query any number of data sources in a transparent manner, as if the client would access one big unified topic map. The architecture uses a layered protocol stack with standardized access methods (TMQL, TMAPI).

Our prototypical implementation in C++ is being used in the ESA funded project SATOPI to provide unified data access to earth observation data (satellite images, glacial sensory data) stored in distributed databases. Using our approach, data collection and preprocessing of earth observation data can be performed by independent systems, whereas all this data is integrated into a unified topic map to enable semantic interpretation and navigation.

However, many crucial aspects such as accessing different data sources, modification of external data, caching, performance and security considerations have been neglected so far. These aspects will be the focus of future work.

7 Acknowledgments

This paper describes part of the work done in the SATOPI project which is being developed by Space Applications Services as a co-funded activity with the European Space Agency (ESA contract number 21520/08/I/OL).

References

1. Barta, R.: Knowledge-Oriented Middleware Using Topic Maps. In: Scaling Topic Maps. Springer-Verlag, Berlin (2008)
2. Garshol, L., M.: TMRAP – Topic Maps Remote Access Protocol. In: Proceedings of the 1st International Workshop on Topic Maps Research and Applications. Springer-Verlag, Berlin (2005)
3. Heuer, L., Schmidt, J.: TMAPI 2.0. In: TMRA 2008, Fourth International Conference on Topic Maps Research and Applications, http://tmapi.org (2008)
4. ISO/IEC 13250-2:2006. Topic Maps – Data Model, http://www.isotopicmaps.org/sam
5. ISO/IEC 13250-6:2008 (draft 2008-05-15). Topic Maps – Compact Syntax (CTM), http://www.isotopicmaps.org/ctm
6. ISO/IEC 18048:2008 (draft 2008-07-15). Topic Maps Query Language, http://www.isotopicmaps.org/tmql
7. Korthaus, A., Henke, S., Aleksy, M., Schader, M.: A Distributed Topic Map Architecture for Enterprise Knowledge Management. In: Proceedings of the 1st IEEE/ACIS Int. Workshop on Component-Based Software Engineering, Software Architecture and Reuse (2006)
8. Korthaus, A., Schader, M.: Using a Topic Grid and Semantic Wikis for Ontology-Based Distributed Knowledge Management in Enterprise Software Development Processes. In: Proceedings of the 10th IEEE International In Enterprise Distributed Object Computing Conference Workshops (2006)
9. Lenzerini, M.: Data integration: a theoretical perspective. In: Proceedings of the 21st ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems (2002)
10. Ontopia. Open-Source Topic Maps engine. http://code.google.com/p/ontopia.
11. TopiEngine. Open-Source Topic Maps engine. http://launchpad.net/topiengine.
12. Wache, H., Vögele, T., Visser, U., Stuckenschmidt, H., Schuster, G., Neumann, H., Hübner, S.: Ontology-Based Integration of Information, A Survey of Existing Approaches. In: Proceedings of the IJCAI-01 Workshop on Ontologies and Information Sharing (2001)