UDC 004.89

USING ONTOLOGY FOR QUERYING IN RELATIONAL DATABASE

O.V. TKANKO, A.I. PETRENKO

Nowadays, the Web is the biggest existing information repository. However, to operate with its information human action is required, but the Semantic Web aims to change this. It provides a common framework that allows data to be shared and reused across application, allowing more uses than the traditional Web. Most of the information on the Web is stored in relational databases and the Semantic Web cannot use such databases. Relational databases can be used to construct ontology as the core of the Semantic Web. This task has attracted the interest of many researches, which have made algorithms (wrappers) able to extract structured syntactic information in an automatic or semi-automatic way. At our work we drew experience from those works. We showed different approaches of formalization of a logic model of relational databases, and a transformation of that model into OWL, a Semantic Web language. We closed this paper by mentioning some problems that have only been lightly touched by database to ontology mapping solutions as well as some aspects that need to be considered by future approaches.

INTRODUCTION

Web is a big pool of information stored in various forms. It requires human operator to perform operations, such as data storage, retrieval and aggregation. But it is possible for a computer to do them without any guidance. The Semantic Web is a project that aims to change that by presenting Web page data in such way that it is understood by computers, enabling machines to do the searching, aggregating and combining of the Web's information. It provides a common framework that allows data to be shared and reused across application, enterprise, and community boundaries [1]. It is a collaborative effort led by W3C with participation from a large number of researchers and industrial partners.

The organization of this paper is as follows: Section 2 describes the concept of Semantic Web and its major layers. Section 3 focuses on converting Relational Schemas to DB Ontologies together with modern semantic query expansion techniques. Section 4 describes existing frameworks and the major interactions between components. Finally, Section 5 concludes the paper by summarizing research and future work.

SEMANTIC WEB CONCEPTS

Semantic Web has layered architecture, which primarily consists of:

- URI and Unicode
- XML: The Representation Layer
- Resource Description Framework (RDF)
- RDF Schema (RDFS) Ontology Layer [2, 3].

Fig. 1 illustrates the architecture of the Semantic Web in a stack manner, where each layer exploits and uses capabilities of the layers below.

A URI is simply Web identifier. In fact, the World Wide Web is such a thing: anything that has a URI is considered to be "on the Web" [4].

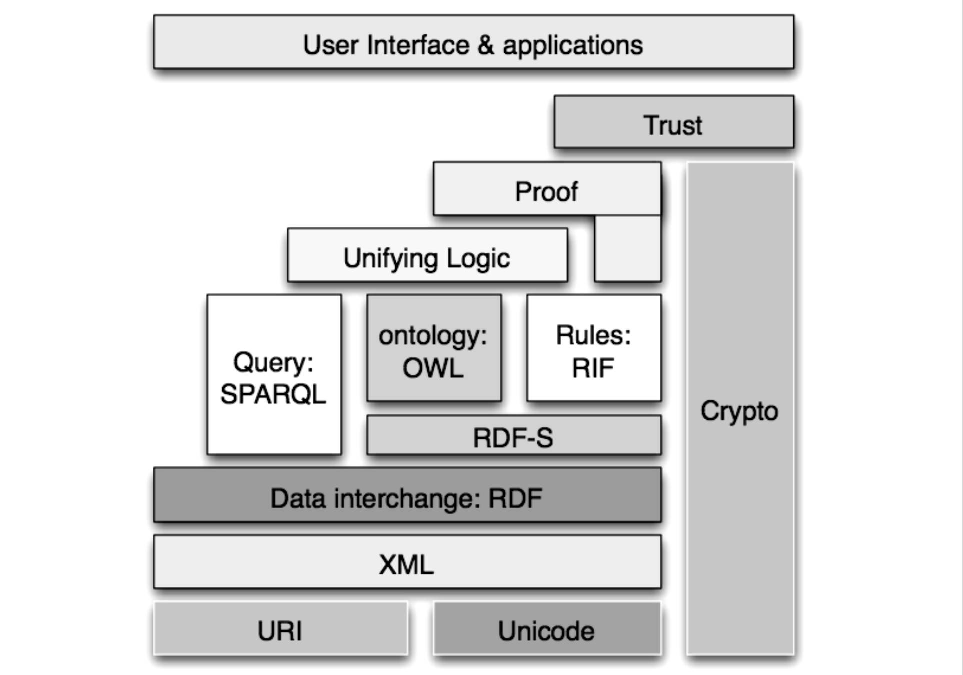


Fig. 1. The Layered Architecture of Semantic Web

The Semantic Web is built on syntaxes which use URIs to represent data, usually in triples based structures: i.e. many triples of URI data that can be held in databases, or interchanged on the world Wide Web using a set of particular syntaxes developed especially for the task. These syntaxes are called "Resource Description Framework" syntaxes.

Unicode serves to represent and manipulate text in many languages. Semantic Web should also help to bridge documents in different human languages, so it should be able to represent them.

XML is a mark up language that enables creation of documents composed of structured data. Semantic web gives meaning (semantics) to structured data.

Middle layers contain technologies standardized by W3C to enable building semantic web applications.

Resource Description Framework (RDF) is a framework for creating statements in a form of so-called triples. It enables to represent information about resources in the form of graph — the semantic web is sometimes called Giant Global Graph.

RDF Schema (RDFS) provides basic vocabulary for RDF. It containsthe definition of:

- classes of individual resources;
- properties, connecting two resources;
- hierarchies of classes;
- hierarchies of properties;
- domain and range constraints on properties.

Web Ontology Language (OWL) extends RDFS by adding more advanced constructs to describe semantics of RDF statements. It allows stating additional constraints, such as for example cardinality, restrictions of values, or characteristics of properties such as transitivity. OWL is based on description logic and so brings reasoning power to the semantic web. Its properties are binary relationships and are distinguished in object and data type properties. Object properties relate two individuals, while datatype properties relate an individual with a literal value.

As with ontology definition languages, more than a few ontology query languages exist, but the de facto query language for RDF graphs is SPARQL. It can be used to query any RDF-based data (i.e., including statements involving RDFS and OWL). Querying language is necessary to retrieve information for semantic web applications. SPARQL uses RDF graphs expressed in Turtle syntax as query patterns and can return as output variable bindings (SELECT queries), RDF graphs (CONSTRUCT and DESCRIBE queries) or yes/no answers (ASK queries).

An alternative way of modeling knowledge is rules, which can sometimes express knowledge that cannot be expressed in OWL. Several ontology languages have been proposed for the implementation of ontologies, but RDF Schema (RDFS) and Web Ontology Language (OWL) are the most prominent ones.

CONVERSION RELATIONAL SCHEMAS INTO DB ONTOLOGIES

Converting available data stored in relational database into RDF format is tedious and it is clearly better if ontology-based queries could directly retrieve the specific data required via SQL rather than first transforming potentially gigabytes of relational data into RDF. It means that integrating existing relational databases with ontology-based systems becoming one of the most important research problems for the Semantic Web.

When Semantic Web agents (which use ontologies) want to interact with relational databases, they need to deal with both the semantic differences between ontologies and schemas and the syntax differences (e.g., OWL vs. SQL).

A relational schema is a finite set $R = (R1, R2, \ldots, R)$ of relations. On the other hand, Semantic Web ontologies (i.e., OWL ontologies) use description logic (i.e., a decidable fragment of first-order logic) as their logic foundation. OWL ontologies mainly have classes, binary predicates (properties) and some axioms (such as cardinality constraints). So underneath of modern semantic query expansion techniques lies idea of automatically creation a Semantic Web ontology which can describe the semantics and structure defined by a database schema, then Semantic Web agents can query the corresponding database based on that Semantic Web ontology.

Semantic query expansion uses low-level heuristics (Fig. 2) between a relational database and its DB ontology listed below and includes syntax wrappers to generate DB ontology from a database schema.

The methodology, proposed by Ayesha Banu, Syeda Sameen Fatima, Khaleel Ur Rahman Khan [5], consists of 2 phases: offline ontology extraction and online query issuing. In offline ontology extraction, the system extracts the explicit classes and relations from the relational schema. Then the domain expert will adapt the extracted ontology by adding the implicit relations to complete the ontology. In online query operation the user can issue a semantic query to the system, and the system maps that query into a related SQL query for the underlining relational database.

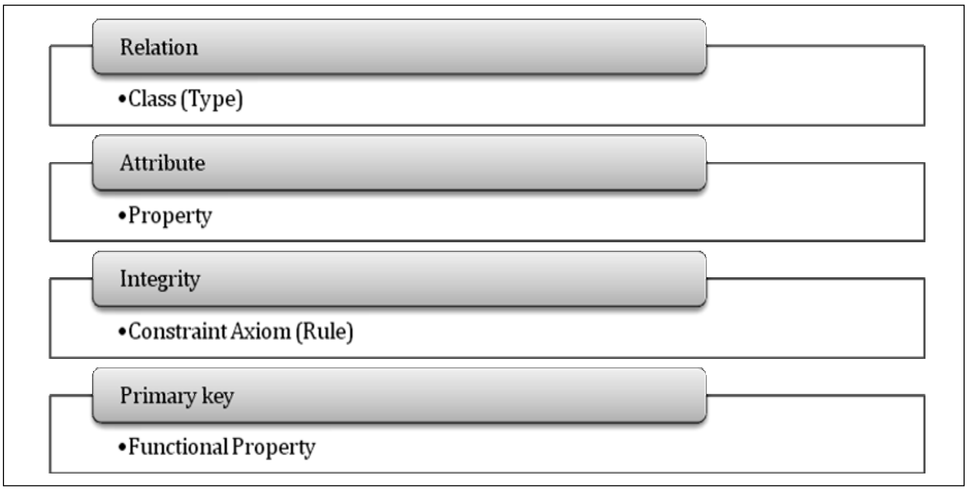


Fig. 2. Low-level heuristics between a relational database and its DB ontology

To extract ontology from Relational Database2main rules are applied:

- if the primary key of any relation is unique and do not contain the primary key of any other relation then we consider such relation as on ontological class;
- if the foreign key of any relation R1 is the Primary key of any other relation R2 then there exists an object property from R1 to R2 and the domain is R1 and range is R2.

Those rules are non-final, because not all object type properties are defined. And in similar work [6] Mostafa E. Saleh enriched those rules with 2 more:

- if the foreign key in a relation R1 is a primary key in another relation R2, then there is an object property (named by its name in R1) from R1 to R2, and the domain is R1, and range is R2;
- if the relation primary key consists of two other primary keys, then that relation is a property between two classes (resources), the classes are the two relations denoted by the two primary keys.

The next step will be adoption of the extracted ontology to pre-defined domain ontology. This stage will add the explicit definition of the implicit relationships and adjust directions of the object properties between classes.

After extracting and refining the wrapper ontology, the end-user issues semantic queries based on extracted ontology concepts and these queries will be mapped onto plain syntactic SQL queries.

SEMANTIC QUERY EXPANSION TECHNIQUES

The term «semantic query» refers to database queries that are based on concepts, properties and instances defined in an ontology and that return semantically relevant results. Approach, presented in [7], defines 3 types of semantically relevant results based on how results are obtained and their relationship to the semantic query.

- *Direct Results* — obtained directly from the database tables. They consist of only results that are explicitly listed in the database tables.
- *Inferred Results* — inferred using the information that is explicitly listed in the database and the domain knowledge in the ontologies. The inference is done using Description Logic reasoning.
- *Related Results* — obtained using data in the database tables and adefinition of similarity of concepts and individuals based on the data model in the ontologies. It includes results that do not strictlymatch the user's query, but may still be similar to the actual answers and hence, may also be semantically relevant.

The end-user issues semantic queries based on ontology concepts and these queries will be mapped onto plain syntactic SQL queries. The semantic queries are based on SPRQL where the user can issue either schema query, or data query. Schema query focuses on querying RDF schemas (ontology) regardless of any underlying instances. Data query is related to semantically navigates/filters instances.

Harris was one of the first to systematically consider SPARQL to SQL translation discussing various ways of organizing RDF triple stores and considers especially using SQL back-ends [8, 9].

The SPARQL query language is based on matching graph pattern. A SPARQL query is defined as on Fig. 3.

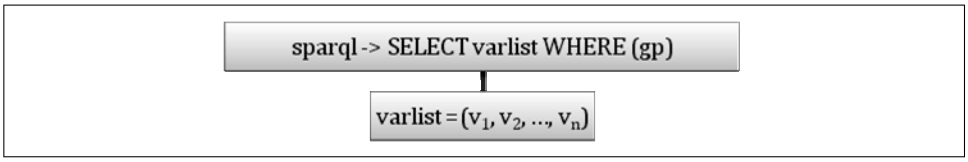

sparql -> SELECT varlist WHERE (gp)
varlist = (v_1, v_2, ..., v_n)

Fig. 3. SPARQL Query pattern

Combining triple patterns gives a basic graph pattern. Combining smaller patterns forms more complex graph patterns. The query generalization algorithm works by repeatedly applying these strategies to generate more and more general queries until a certain pre-specified number of results are obtained [14].

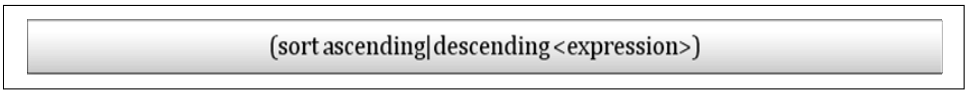
Lately, Chebotko presented a method for translating a SPARQL query to a single SQL query with preservation of semantics [10]. Their method operates on SPARQL algebraic level, and relies on SQL sub-queries on data set declaration.

A rather flexible translation approach is presented in [11]. The procedure is defined as follow:

- The triples that share the same subject are grouped as they represent the same table information. So, each group represents some information about one concept in mediated ontology.
- Based on the mapping information, the translation algorithm replaces all predicates in the triples with corresponding columns name in relational databases tables.
- If the predicate is not in the columns name, then it will be in object property names related to the linking tables.
- For each separate group, a sub-query clause is created, which consists of three parts: SELECT, FROM and WHERE clauses. The SELECT clause is cre-

ated according variable occurs both in triple and in SPARQL select clause. The FROM clause is created according the column name in the triples. And the WHERE clause is created according the columns and mapping information. After all clauses are created, we can combine them and construct the complete a query clause.

After executing the SQL query against relational database the result set is acquired. After the result data set is formed, order expressions are used to sort the data set. The first order expression is evaluated for each row and then the rows are ordered so that the rows with smaller order value are enumerated first in ascending order or vice versa in descending order. If two or more rows have the same order value, the second order expression is used to determine the ordering between these rows, and so on. The sort expressions presented in Fig. 4.



(sort ascending|descending<expression>)

Fig. 4. Sort expressions

On the final stage formatting algorithm transforms the result sets from relational database into RDF triples using the namespace and URIs.

MODERN ONTOLOGY-BASED FRAMEWORKS

There are several major ontology-based frameworks that provide a unified semantics for mapping discovery and query translation by transforming database schemas to Semantic Web ontologies. We found as the most attractive:

- OntoGrate — ontology-based framework that provides a unified semantics for mapping discovery and query translation by transforming database schemas to Semantic Web ontologies.
- Cross — an OWL Wrapper for Reasoning on Relational Databases.
- Jena based frameworkprovides the RDF data sources and querying.
- Ultrawrap — automatic tool that automatically exposes relational databases as RDF and allows them to be queried using SPARQL.

The OntoGrate [12] system can automatically represent a schema as a DB ontology. With the generated DB ontology, a semantic web query (e.g., in OWL-QL) can be directly translated into a SQL query and the answers (relational data) can be translated back to semantic web languages (e.g., RDF and OWL).

The system is mainly composed of five components: the ontology matching, the rule miner, the inference engine, the query interface, and syntax wrappers. The transformation from schema to ontology is implemented in the wrappers between SQL and OWL. The inputs are relational databases and Semantic Web documents and queries over heterogeneous schemas or ontologies from various domains. In between Onto Grate and the data resources exist syntax translators (wrappers) among OWL, SQL, OWL-QL and Web-PDDL. The query wrapper takes fully translated ontology-based query and efficiently generates the corresponding data access SQL query without additional translation or rewriting cost. Until the proposed standards for semantic mappings are finalized, Web-PDDL is used internally to describe both the structure and semantics of data resources, their mappings and queries.

There are several differences between the theoretical model described above and the actual implementation. At first there is an effort to reduce the redundancy in the OWL knowledge base. In Section 3 we noticed that transformation ψ creates information by associating to every uniqueness constraint a property, which is independent of the properties, associated to the columns concerned by that constraint. And it is redundant. This is useful for multi-column constraints to guarantee the uniqueness can only apply to a single property. This is what Cross-does, and it does the same for properties representing foreign keys. The second difference is that, for the sake of completeness, the implemented transformation includes an axiom forcing foreign keys to point to an existing value. The third difference is about the representation of data. While the transformation straightforwardly creates an individual per row and an individual per data value, Cross introduces an intermediate layer of individuals. Cross's preliminary results are encouraging: the transformation of the schema of real database (127 tables, 869 columns, 132 unity constraints, no foreign key) took around 1.5s; the resulting ontology was loaded in Pellet in about 9s, while reasoning took about 3s.

Ultrawrap [13] automatically exposes relational databases as RDF and allows them to be queried using SPARQL. OWL ontology is generated and then it can be mapped to domain OWL ontology through a GUI. This tool makes maximal re-use of existing commercial SQL infrastructure by letting the SQL optimizer do the SPARQL query execution. A purely automated procedure would need to make oversimplifying assumptions on the lexical proximity of corresponding element names in the database schema and the ontology that are not always true. Such assumptions tend to overestimate the overall efficiency of mapping discovery methods.

CONCLUSIONS AND FUTURE WORK

Relational databases are considered one of the most popular storage solutions for various kinds of data. In this paper, we described existing methods for making them accessible by the Semantic Web.

First, relational data can be physically converted to RDF and then stored in a RDF triple store. An advantage of this approach is that it is a straightforward and fast in achievement data integration. The disadvantage is clear — creation of a separate copy of the relational data. Furthermore, there is dependency on the existing RDF triple.

A different approach is not to materialize the relational data as RDF and leave it in the relational database. Creating a mapping between the relational data and RDF can allow "on-the-fly" SPARQL queries on top of a relational database. This approach enables a SPARQL query to execute over different data sources by following the links between RDF data on the web.

The mapping process of the semantic query into SQL statements involves many aspects. We outlined rules that are applied in popular modern frameworks together with translation algorithm described in [1]. Onto Grate [12] uses an approach of deductive query answering, which rewrites original queries into a finite set of conjunctive queries in terms of the DB schema. A big advantage is the usage of hetero generous databases in a highly automatic way. As the result subquery clause is created, which consists of three major parts: SELECT, FROM and

WHERE. Implementation of a client-server API will be a comprehensive effort. Primary there is a need to get more experimental results for all implementations together with investigation of behave or in large-size relational databases of multiple domains.

A semantic querying relational database has a very exciting future in the short term. One of the interesting challenges in the long term is to see the adoption of the standard by the major database vendors.

REFERENCES

1. *W3C* Standards. — http://www.w3.org/RDF/FAQ.
2. *OWL* Web Ontology Language. — http://www.w3.org/TR/owl-ref/.
3. *Jun Cai, Vladimir Eske, Xueqiang Wang*. Semantic Web & Ontologies. — http://www.mpi-inf.mpg.de/departments/d5/teaching/ss03/xml-seminar/talks/CaiEskeWang.pdf.
4. *The Semantic* Web: An Introduction. — http://infomesh.net/2001/swintro/#whatIsSw.
5. *Ayesha Banu*. Semantic — Based Querying Using Ontology in Relational Database of Library Management System / Ayesha Banu, Syeda Sameen Fatima, Khaleel Ur Rahman Khan // Canadian Journal on Data, Information and Knowledge Engineering. — 2011. — **2**, № 1. — P. 21–28.
6. *Mostafa E. Saleh*. Semantic-Based Query in Relational Database Using Ontology // Canadian Journal on Data, Information and Knowledge Engineering. — 2011. — **2**, № 1. — P. 1–16.
7. *Anand Ranganathan*, *Zhen Liu*. Information Retrieval from Relational Databases using Semantic Queries. — http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.87.5718&rep=rep1&type=pdf.
8. *Sami Kiminki*. SPARQL to SQL Translation Based on an Intermediate Query Language. — http://www.cs.hut.fi/~skiminki/bib/ssws2010-paper3.pdf.
9. *Harris S*., *Shadbolt N*. SPARQL query processing with conventional relational database systems. Date Query Language. — 2005. — **3807**. — P. 235–244.
10. *Chebotko A*., *Lu S*., *Fotouhi F*. Semantics preserving SPARQL-to-SQL translation // Data and Knowledge Engineering. — 2009. — **68**, № 10. — P. 32–39.
11. *Riazanov A*. Resolution-based Query Answering for Semantic Access to Relational Databases // A Research Note. Preprint. — 259, 48 — P. 1–15.
12. *Lubyte L*. Tessaris S. Extracting ontologies from relational databases // Proceedings of Description Logics. — 2007. — **250**, № 48. — P. 122–126.
13. *Sequeda J.F.* Ultrawrap: Using SQL Views for RDB2RDF. — http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.153.6203&rep=rep1&type=pdf.
14. *Dejing D., Han Qin, Paea LePendu*. Ontograte: towards Automatic Integration for Relational Databases and the Semantic Web through an Ontology-Based Framework // Int. J. Semantic Computing. — 2010. — **4**, № 1. — P. 123–151.

Received 02.04.2013

From the Editorial Board: the article corresponds completely to submitted manuscript.