Analytical Review of Data Transformation for the Task of Integrating Various Representations on the Example of Ontologies and Relational Databases

Nadezhda Yarushkina[1[0000-0002-5718-8732]] and Vladislav Moiseev[2[0000-0003-2671-4653]]

[1] Ulyanovsk State Technical University, Ulyanovsk, Russia
jng@ulstu.ru
[2] Ulyanovsk State Technical University, Ulyanovsk, Russia
vladdy.moses@gmail.com

Abstract. This article provides an overview of the data transformation solutions needed to integrate various models and data schemas. An example of mutual mapping of ontologies on relational databases is considered. The scientific basis of the presented data models is described. Also set goals and objectives for data integration. The article describes the problems and further steps to explore the possibility of data transformation in computer-aided design systems. The main problem of transforming a relational database into an ontology is the lack of metadata. With the inverse transform, some data and metadata may be lost. In addition, there are several approaches to the transformation of ontologies into relational databases. OWL is used as the ontology description language. The article is useful to architects and software designers, as well as knowledge engineering specialists.

Keywords: Data Transformation, Data Integration, Ontology, Relational Database, OWL, ALC.

1 Introduction

The present stage of human development is characterized by an enormous speed of development. At the same time, science, originally formed as a practical application of thought, has long been theoretical and is increasingly divorced from practice, and therefore also technology. However, no technological task is possible without a scientific basis.

In particular, such a merger and separation of science and technology is observed in the description of our world, namely in the representations of knowledge and data.

The most famous data access technology, invented in the 20th century, is the relational database and the language for managing and manipulating SQL. This technology is widely used due to high speed, reliability and unification of access. Unlike many modern programming languages, the syntax of SQL, firstly, does not indicate specific commands for obtaining data and, secondly, is practically independent of the specific implementation. It is also important to note that the speed of data manipulation was

Proceedings of the XXII International Conference "Enterprise Engineering and Knowledge Management" April 25-26, 2019, Moscow, Russia

achieved thanks to the mathematical apparatus of relational algebra or relational calculus. Many modern non-relational DBMSs (also called NoSQL) do not have the same coherent formal description, therefore, they are not as widely used as SQL.

Another approach to data presentation are ontologies with a scientific basis in the form of ontological analysis and descriptive logic. It was thanks to the descriptive logic that it was possible to express in a formal form, and then implement all sorts of dictionaries, as well as identify contradictions in the concepts of ontologies through reasoners. The practical application of ontologies is the formation of thesauruses, as well as an attempt to present the meaning on the world wide web (semantic web). The last application is especially important in the world of the Internet of things and search engines.

However, it is worth noting that although relational databases and ontologies allow knowledge to be presented with some degree of similarity, these approaches developed in parallel and do not find many points of contact precisely because RDBMS is the development of technology, and ontologies are the development of science.

We should also consider the issue of the documentability of modern information systems. To summarize, the documentation is a representation of the work of the system in a form convenient for a person. That is, it is a linguistic mapping of data models and business processes described within the system. At the same time, an ontology or a relational database (the latter, however, with some limitations) can be used to describe the subject domain.

It turns out that there are at least three important and different descriptions of the domain: a relational representation (technological), an ontological representation (scientific), and documentation (linguistic). And all representations denote a single whole, but in different ways. Therefore, all data models must be consistent with each other, or, in other words, be expressed through each other or through some other knowledge representation model.

Chapter 2 analyzes the various schemas and data representations to select the main schema through which others can be presented. The third chapter deals with the problem of data transformation using the example of ontologies (OWL technology) and relational databases. The fourth chapter describes the theoretical foundations of data transformation. The conclusion describes the results of the analytical review and describes the subsequent course of research in the current field.

2 Selection of the Main Data Schema

When there are several representations (schemas) of data, it is necessary to choose one through which all others can be expressed.

The concept of data integration between various schemes is described in [5], where the following key concepts are used to describe the integration mechanism:

- schema / query - data presentation in a specific case;
- global scheme - a representation common to all data schemes;
- local scheme or source scheme - the data model used in a specific task;
- mapping - tables of data transfer from one scheme to another.

Also, in this paper several data conversion processes are considered: LAV (local as view) and GAV (global as view).

The LAV approach is based on the idea that the concepts of each source scheme should be described within the global scheme. This is possible if the global scheme is stable and well represented. An ontology can act as a global scheme. Also, this approach makes it easy to expand the system by adding new sources. It is only necessary to describe their model in global terms.

The GAV approach, on the contrary, states that the concepts of a global scheme should be expressed in terms of sources. This is only possible if the sources are stable. In other words, mapping with this approach explicitly states how and where to get data in terms of a global scheme. This provides for a declarative approach, since when manipulating data, we do not explicitly indicate where to get them, but only refer to the global schema.

Both approaches can be used in building data integration between the schemas. In addition, both approaches can be transformed if desired, which is also described in [5].

In our case, the LAV approach will be chosen, since it has a stable core - a global scheme. Further, ontology with justification in the form of description logic will be considered as such a scheme.

3 The Problem of Transformation of Data Models

With the modern development of science and technology, there are many data models that can provide the same information about the real world in various forms.

The most interesting for the task of data transformation are relational database management systems and ontologies, since both models have a scientific basis: relational algebra or relational calculus in a RDBMS, and display logic in ontologies.

3.1 Transformation of RDBMS into OWL Ontology

There are many strategies for transforming relational schemes into ontology. For example, one can single out the concept of direct mapping from [15,1]. It consists of describing concrete examples of mapping relational data structures to the RDF (Resource Description Framework) metadata description technology developed by the W3C. A formal description of direct mapping is also given. Also, within the framework of this transformation model, the possibility of fine-tuning the initial and resulting data set is excluded. Therein lies the “directness” of the comparison — in the exact mapping.

Another technology for transforming a relational scheme into metadata, also developed by W3C, is the R2RML language [16]. Unlike direct mapping, this scheme allows you to more deeply adjust the individual parts of the transformation. The W3C recommendation does not contain a formal description of the R2RML transformation language, but it contains many specific examples of data transformation.

There are also scientific papers describing the transformation of relational data schemas into ontologies, and not only into RDF metadata. For example, a comprehensive review of current developments in such a transformation is given in [2]. Also in this article highlights the main types of entities and the basic rules of conversion, describes its own data conversion.

Based on [2], to consider the issue of data transformation, the following types of entities can be distinguished:

- Strong entities without foreign keys (also known as directories);
- Strong entities with foreign keys;
- Weak Entities:
  - With a matching foreign and primary key;
  - With a composite primary key containing various foreign keys to all fields;
  - With a composite primary key, while the attributes of the original entity are not duplicated in the reference (referenced);
  - With a composite primary key, with some simple attributes may be duplicated with reference entities.

The categories described above are highlighted to correctly define the transformation rules. For example, entities without foreign keys can be transformed into ordinary ontology classes.

It is also necessary to consider the limitations of drying. For example, the constraint constraint (NOT NULL) is set using MinCardinality (in OWL notation), and the uniqueness constraint (UNIQUE) is set using the InverseFunctionalProperty property [2].

Some transformation algorithms from RDBMS to ontologies (for example, in [3]) offer a gradual transition: classification of tables, their comparison, comparison of columns on data properties (data property), comparison of relations and comparison of restrictions.

When constructing matching rules, it is worth bearing in mind the correctness (in other words, relationality) of the design of the original data scheme. There must be foreign keys between entities, since without them, ontology will degenerate into a set of concepts without any connections. It is also important to give descriptions and literate restrictions (at least on compulsion and on uniqueness) in the tables, since without this, it will be impossible to build inference systems in ontology.

3.2 Approaches to the Transformation of Ontologies in RDBMS

The inverse approach to translating schemas is to translate from an ontology scheme into a relational data model.

The reviews [4, 12] describe the transformation classes of ontologies OWL into relational data schemas. The main approaches described in the article are:

1. Using the conceptual graph;
2. Saving facts in a relational structure so that you can execute SQL queries on the data presented;
3. Transformation based on predefined rules for the transfer of ontology to a relational structure;
4. The ontology representation in the form of a graph stored in a relational database;

5. For ontologies with fuzzy data types, these types are also described in a relational database in a separate schema;
6. Ontology virtualization for applications (in fact, is not a transformation, an API is provided for applications by analogy with relational databases).

The main problems here are either limitations of the approach (for example, support only of the basic constructs of the OWL language), or initial tuning of the ontology, or poor quality of the resulting relational data schema.

4 Theoretical Foundations of Transformation

The basis of any technology is science.

The basis of ontologies is descriptive logic. They represent a family of logical formalisms based on the concepts of the concept (class of concepts), the role (relationship) and the individual [17]. Relationships of descriptive logic with decidable fragments of first-order logic (predicate calculus) are also mentioned. In [18], it is mentioned that descriptive logics are fragments of first-order logic.

The scientific basis of relational databases is relational calculus or relational algebra. According to Codd's theorem, query languages for relational data, relational algebra and relational calculus are equally expressive [19]. This means that any query (predicate calculus) and any relational data structure (tuple calculus) can be represented as a well-constructed first-order logic formula.

However, in the case of comparison of description logic and relational algebra, it is necessary to compile a terms comparison table. It is presented in table 1.

Table 1. Terms comparison table between description logic and relational algebra.

Term of description logic	Term of relational algebra
Concept	Relation
Role	Foreign key
Individual	Tuple
Property	Attribute
Metadata	Constraints

For example, let *Person* be the concept denoting a person, *Female* be the concept "female being", Male the concept "male being". Also declared the role of *hasChild*. Then the concept of *Person* ⊓ *Female* will designate a woman, and the concentrated *Person* ⊓ ∃*hasChild.Person* will designate a person who has a child. Examples are taken from [17]. In the relational data model for such a transformation, we can declare the relationship "*Man*" with a set of attributes:

- person_id - person identifier;
- is_male - is a male creature;
- is_female - is a female being.

Additionally, you will need to enter the *PersonChild* entity with the following attributes:

- person_id - person identifier;
- child_person_id is the ID of the person-child.

Thus, in order to obtain all women, it is necessary to apply the relational sampling operation (restrictions) with *is_female = true*. To select all parents, it is necessary to relation *Person* relational join operation with the *PersonChild* relation predicate *Person.person_id = PersonChild.person_id*.

Also, in this chapter, it is necessary to separately note the work [6], in which an attempt was made to map the descriptive logic of ALC to the relational model (special relational structure RM2). The paper presents evidence for mapping ALC to the relational model RM2. However, the questions of reverse mapping and the use of other relational models and descriptive logics remained unresolved.

5 Conclusion and the Following Work

At the current time, the transformation of data models is widely used in the construction of complex information systems. This is due to the incompatibility of different technologies with each other, as well as due to duplication of the description of the same subject area several times, for example, in the description of the relational data schema, in the ontology of the subject area, in artifacts of the information system (documentation).

To reduce the cost of maintaining various data models that describe the same subject area, it is necessary to correctly and reasonably convert data schemas.

At the current time, data transformation from both RDBMS to ontology (usually using OWL and the Protege editor) and reverse mapping are considered. However, the following tasks are poorly addressed:

- description and implementation of ontology transformation into a relational structure using DBMS metadata (for example, names of foreign keys, unique constraints, description of tables and fields);
- a description of the above transformations from a formal point of view;
- proof of the isomorphism of the descriptive logic and relational calculus

These questions should be considered in the following papers.

References

1. Sequeda1 J., Arenas M., Miranker D.: A Completely Automatic Direct Mapping of Relational Databases to RDF and OWL, https://www.researchgate.net/publication/267232319_A_Completely_Automatic_Direct_Mapping_of_Relational_Databases_to_RDF_and_OWL.
2. Louhdi M., Behja H., Alaoui S.: Transformation Rules For Building OWL Ontologies from Relational Databases, https://airccj.org/CSCP/vol3/csit3822.pdf.

3. Astrova A., Kalja A.: Mapping of SQL Relational Schemata to OWL Ontologies, http://wseas.us/e-library/conferences/2006elounda1/papers/537-193.pdf.
4. Vysniauskas E., Nemuraite L.: Mapping of OWL Ontology Concepts to RDB Schemas, http://isd.ktu.lt/it2009/material/Proceedings/OCM.pdf.
5. Lenzerini M.: Data Integration: A Theoretical Perspective, https://www.researchgate.net/publication/220266329_Data_Integration_A_Theoretical_Perspective.
6. Reznichenko V., Chistyakova I.: Mapping ALC Descriptive Logic to a Binary Relational Data Structure [Publication in Russian], https://core.ac.uk/download/pdf/87412875.pdf.
7. Kolchin A., Eliseeva N.: Presentation of the Knowledge Model of the Specialist-Designer on the Basis of the Ontological Approach [Publication in Russian], https://elibrary.ru/item.asp?id=11909222.
8. Ivashko A., Grigoriev A., Kropotin A.: Proof of Equivalence Object Relational Mapping [Publication in Russian], https://elibrary.ru/item.asp?id=25064604.
9. Kropotin A.: Application of Description Logic Formalism to Identify Semantic Conflicts of Conceptual Schemes Entity-Relationship [Publication in Russian], https://elibrary.ru/item.asp?id=28116338.
10. Dragalina-Chernaya E.: Formal Ontologies as Abstract Logics [Publication in Russian], https://iphras.ru/uplfile/logic/log12/Li_12_dragalina.pdf.
11. Bumans G.: Mapping between Relational Databases and OWL Ontologies: an Example, https://www.lu.lv/materiali/apgads/raksti/756_pp_99-117.pdf.
12. Humaira A., Tabbasum A., Ayesha S.: A Survey on Automatic Mapping of Ontology to Relational Database Schema, https://pdfs.semanticscholar.org/ef1d/017a9f8f99961b5a526b922e5a7a2cbe2146.pdf.
13. Vysniauskas E., Nemuraite L.: Transforming Ontology Representation from OWL to Relational Database, http://donelaitis.vdu.lt/publications/Vysniauskas_2006.pdf.
14. J. Barwise, S. Feferman, eds., Model-Theoretic Logics (New York: Springer-Verlag, 1985), 3-23, https://projecteuclid.org/euclid.pl/1235417266.
15. A Direct Mapping of Relational Data to RDF, http://www.w3.org/TR/rdb-direct-mapping/.
16. R2RML: RDB to RDF Mapping Language, https://www.w3.org/TR/r2rml/#dfn-r2rml-mapping-document.
17. Konev B.: Ontologies and Knowledge Representation - Description Logics [Publication in Russian], https://compsciclub.ru/media/slides/ontology_2010_autumn/2010_09_25_ontology_2010_autumn_EaSzuoZ.pdf.
18. Kikot S.: Description Logics and Their Complexity [Publication in Russian], http://ru.discrete-mathematics.org/?page_id=2221.
19. Pichler R.: Database Theory - Codd's Theorem, http://www.dbai.tuwien.ac.at/staff/pichler/dbt/slides/dbt03.pdf.