REALIZATION OF THE ONTOLOGICALLY BASED METHOD FOR CHECKING STRUCTURAL INCONSISTENCES OF RELATIONAL DATABASES

Alexander A. Kropotin, Andrei V. Grigoryev, Yuliya V. Bidulya, Alexander G. Ivashko & Nikita S. Durynin

Abstract

In paper the problem of use of technologies of a semantic web for automatic detection of structural inconsistencies of relational databases in case of software integration is solved. The description of the conflicts of relational databases in case of their integration is provided in chapter 1, methods of their determination and reasons for development of own method of the solution of an objective. The description of a method of automatic detection of structural mismatches of relational databases with use of technologies of a semantic web is provided in chapter 2. The description of experimental evaluating of the developed method is provided in a chapter 3 and its results are represented. In the inference plans for future development in this area are defined.

Keywords: Ontology; description logics; database; ER-model.

1. Introduction

At present the integration of information systems and their databases is one of the popular research task [1, 2, 3]. Data integration may be formally defined like synchronization of data between databases of the software which can be provided by different data schemes in different sources.

A lot of scientific schools around the world take part in researches related to information systems integration, such like: Database Technology University of Zurich [1], Wright State University [2], Dublin City University School of Computing [3] and other. Besides, there are different methods and approaches to integration of data [4, 5, 6, 7], but all of them face with the problems of integration of data connected to a structural mismatch of data schemes [4], which may be defined as conflicts related to

a) schemes inconsistencies, it may arise in solving problems of integrating schemes;
b) data inconsistencies, it may arise in solving problems of integrating schemes.

The following data schemes differences may be associated with schemes inconsistencies [4, 5]:

2. Method description

We propose the OWL DL implementation to detect problems during the process of data schemes integration. The developed method includes three steps:

a) at the first step, each conceptual schema of relational database must be mapped into ontology.

b) at the second step these ontologies have to be merged into one.

c) at the last step the reasoning for this ontology have to be implemented.

Special for realization of the method steps were developed three models:

a) the ontological model of a conceptual domain objects representation has been developed special for realization of the method first step. The model is is presented as a rules for describing database concepts into description logics terms. The model basedon the expressive description logic SROIQ.

b) the model of ontologies union of an conceptual domain objects has been developed special for realization of the method second step.

c) the model analysis of an united ontology has been developed special for realization of the method last step.

The detailed description of the developed models has been submitted in [22] therefore further, we will provide their short description in the form of rules of the mapping binary entity-relationship diagram elements into ontology elements.

2.1. The mapping conceptual schema of relational database into ontology

Following we present some rules for mapping of semantic network "binary entity-relationship" models to OWL DL ontology. The set of terminological axioms and assertional axioms is proposed to term the conceptual domain objects ontology (1).
\[ \mathcal{K} = \mathcal{T} \cup \mathcal{A}, \quad (1) \]

where \( \mathcal{K} \) is the conceptual domain objects ontology; \( \mathcal{T} \) is a terminological axioms; \( \mathcal{A} \) is an assertional axioms.

**Entity set and value set.** Both entities sets and values sets should be mapped into class. While transforming entities sets and values sets, it should be taken care of tagging classes names to correspond to entities sets names and values sets names. Each class should be made as disjoint with other.

**Entity and value.** Both entities and values would be mapped into individuals. While transforming entities and values, it should be taken care of making local unique names. Each individual should be maked as different with other in frame of one class, when it putted in line with entity.

**Relationship set and attribute.** Both relationships sets and attributes should be mapped into object properties. While transforming relationships sets and attributes, it should be taken care of tagging object properties names to correspond to relationships sets and attributes names.

While transforming attributes, in addition to object properties datatype properties should be created. Thus, each attribute should be mapped into composition of object property and datatype property. While transforming attributes, it should be taken care of making local unique names.

Each object property and datatype property should be made as functional, otherwise OWL DL allows object property and datatype property to take many values by default.

While transforming relationship sets cardinality of «one-to-many», in addition for object properties would be created and tagged as inverse object properties. While transforming relationship sets cardinality of «one-to-one», object properties would be tagged as symmetric. Thus relationship sets would be mapped into two inverse object property or one symmetric object property for each.

The domain and the range of each object property is the class. The domain of each data property is the class and range is the actual datatype (int, string, etc).

**Primary key and unique constraint.** Each datatype property and object property, when it putted in line with primary key attribute, should be tagged as hasKey of their class.

### 2.2. The merging rules of ontologies conceptual domain objects

The merging ontologies domain conceptual objects (1) is proposed to term ontology of set of domain conceptual object different representation (2).

\[ \mathcal{K} = \bigcup_{i=1}^{L} \mathcal{K}_i, \quad (2) \]

where \( \mathcal{K} \) is the ontology of set of domain conceptual object different representation; \( L \) is the count of different ontology union \( \mathcal{K}_i: L \geq 2; \mathcal{K}_i \) is the domain conceptual objects ontology.

Then in this case in integrated terminological axiom set \( \mathcal{T} \subseteq \mathcal{K} \) should be defined equal atomic concepts (1) of different ontologies in terminological axiom terms of a concepts equivalence (3).

\[ \{\{R^1_l \equiv R^h_l\}_{l=1}^{L}| l \in chsl; l = 1L, \quad (3) \]

where \( R^1_l, R^h_l \) are object properties and datatype properties defined on the ontology \( \mathcal{K} \) and \( \mathcal{K}_h \) respectively; \( l^1 \)is a count of all object properties and datatype properties defined on the ontology \( \mathcal{K}_i; L \) is a count of all ontologies \( \mathcal{K}_i \).

In addition, in integrated terminological axiom set \( \mathcal{T} \subseteq \mathcal{K} \) equal classes (4) of different ontologies should be defined in terminological axiom terms of a concepts equivalence (4), (a) if the count of ontology \( \mathcal{K}_i \) object properties is greater than count of ontology \( \mathcal{K}_h \) or (b) otherwise.

\[ \left\{ \begin{array}{l} \{\{A^1_n \equiv A^h_n\}_{n=1}^{N_1}| n \in chsl; n = 1N_1, \quad (a) \\
\{\{A^1_i \prod_{i=1}^l 3S^1.C^1 \equiv A^h_i\}_{n=1}^{N_1}| n \in chsl; n = 1N_1 \} \end{array} \right\}, \quad (4) \]

where \( A^1_n, A^h_n \) are classes of the ontologies \( \mathcal{K}_i \) and \( \mathcal{K}_h \) respectively; \( S^1 \) is the transitiv object property of the ontology \( \mathcal{K}_i; l \) is a count of all properties of the concept \( A^1_n \) are putted in line with properties of the concept \( A^h_n \) which have not equal roles from the ontology \( \mathcal{K}_i; N_1 \) is a count of all classes of the ontology \( \mathcal{K}_i; L \) is a count of the all ontologies \( \mathcal{K}_i \).

In case of integrated assertional axioms the set \( \mathcal{A} \subseteq \mathcal{K} \) contains at least two assertional axioms sets \( \mathcal{A}_i \subseteq \mathcal{A}; \mathcal{A}_i \neq \emptyset \), in which individuals were defined, in these assertional axioms set the individuals equivalence of different ontologies \( \mathcal{K}_i \) should be defined in assertional axioms terms (5).

\[ \{\{a^1_l \equiv a^h_l\}_{l=1}^{L}| l \in chsl; l = 1L, \quad (5) \]
where $a_j^l$, $a_j^h$ are individuals of the ontologies $\mathcal{K}_l$ и $\mathcal{K}_h$ respectively; $J^l$ is a count of all individuals of the ontology $\mathcal{K}_l$; $L$ is a count of all ontologies $\mathcal{K}_l$.

2.3. The checking structural inconsistencies of relational databases

We propose to implement the description logics calculus [8, 23, 24] to ontology of set of domain conceptual object different representation for checking structural inconsistencies of relational databases. Thus it is necessary to resolve three algorithmic problems of description logics calculus, such that:

a) the inconsistence algorithmic problem of an ontology terminological (only Tbox);

b) the classification algorithmic problem of an ontology terminological (only Tbox);

c) the inconsistence algorithmic problem of an ontology (Tbox and Abox);

The resolving of inconsistence algorithmic problem of an ontology terminological could be able to follows detecting relational databases integration problems [4, 5, 6, 7, 8]:

a) difference in the cardinality of relationships set the domain and/or domain range do not overlap;

b) difference of the restrictions value sets range of roles (concrete domain) do not intersect;

c) difference of the types of data range of roles (concrete domain) do not intersect;

d) difference of the set of valid values range of roles (concrete domain) do not intersect.

The resolving of classification algorithmic problem of the terminological ontology could be able to follow detecting semantic network "binary entity-relationship" models integration problems:

a) an absence of elements of schemes 1) if there is no composition of atomic roles, then transitive role is not completed in the place of the absent, and one concept is embedded in the another, but about another we could allege this, so they are not equivalent; 2) absence of equivalent concepts from concepts of one of terminology;

b) difference of «entities sets ¬ value sets» equivalent concepts belong simultaneously to concepts EntitySet and ValueSet;

c) difference of the «value sets ¬ set value sets» 1) equivalent concepts has no concepts in a one terminology; 2) one concept in one terminology has many equivalent concepts in the other;

The resolving of inconsistence algorithmic problem of an ontology could be able to follows detecting relational databases integration problems:

a) difference in the modality of relationships set null value is defined as {nominal} implication for all Role with not a Concept;

b) difference values equivalent individuals belong to equivalent roles with different values.

3. Evaluating

Software implementing presented method was developed. It get database descriptions from Papyrus XML file and transforms it to knowledge bases. Software was developed using Java language, because of there is a library called OWL API that allows creating knowledge bases in OWL language. Besides, there are many open source reasoning software allowing solving task of logical reasoning. To check same names of the tables we use greedy algorithm with similarity checking. In the future researches we consider to use a method based on the Kun algorithm for max weighted matching in bipartite graph.

HermiT reasoner is using as reasoning tool, because of HermiT shows great results on past OWL Reasoning Evaluation Workshop [24]. Unfortunately, we test our software only during development on test databases with no more than 10 tables and tests shows 100% regular results (table 1).

<table>
<thead>
<tr>
<th>Distinction title</th>
<th>Count of structural distinctions</th>
<th>The method result</th>
</tr>
</thead>
<tbody>
<tr>
<td>distinction &quot;entity set – value set&quot;</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>distinction &quot;value set – set of value set&quot;</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>distinction in a set of admissible values</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>distinction in data type</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>distinction in semantics of communications</td>
<td>5</td>
<td>5</td>
</tr>
</tbody>
</table>

Tab. 1. The method evaluating result.
4. Conclusion

This paper describes a problem of automatic checking structural inconsistencies of relational databases. This paper presents a framework for the detecting relational databases integration problems. We have provided rules to transform relational databases concepts into equivalent OWL ontology for semantic web.

This proposed framework helps software engineers in upgrading the structured analysis and design relational databases. Our ongoing research on this topic is to handle other cases of relationships that are not binary, which require reification.

Software implementing presented method was developed. It get database descriptions from Papyrus XML file and transforms it to knowledge bases. HermiT reasoner is using as reasoning tool. Software was developed using Java language and OWL API.

Unfortunately, we test our software only during development on test databases with no more than 10 tables and tests shows 100% regular results.

In the future, we will extend our researches, we plan to develop, and present ready software tested on enterprise databases. Moreover, there are few systems with opened database creating scripts, and to build knowledge bases we will use data description using data definition in web services. For example, WSDL-definition with a XSD schemas can be considered as a source of the database schema.

5. Acknowledgements

We would like to thank our colleagues who have helped in the development of independent parts of the software, but who wish to remain outside the scope of the co-authors of the article: Shaykhullina Irina, Michael Samoilov, Alexander Stupnikov and Dusso Bruno Jean-Marie.

6. References


