

*I.S. Chystiakova*

## IMPLEMENTATION OF MAPPINGS BETWEEN THE DESCRIPTION LOGIC AND THE BINARY RELATIONAL DATA MODEL ON THE RDF LEVEL

This paper is dedicated to the data integration problem. In article the task of practical implementation of mappings between description logic and a binary relational data model is discussed. This method was formulated earlier at a theoretical level. A practical technique to test mapping engines using RDF is provided in the current paper. The mappings DL ALC and its main extensions to the RDF triplets are described in the publication. The mapping of the DL axioms into an RDF triplet also is considered in the publication. The main difficulties in describing DL-to-RDF transformations are given in the corresponding section. The paper also provides an overview of existing methods that relate to the use of RDF when mapping RDB to ontology and vice versa.

Key words: binary relational data model, description logic, mapping, RDF, DL,  $RM^2$ , ALC, OWL.

### Introduction

The research series [1–7] is dedicated to the analysis and solution of the problem of creating a mapping mechanism between the description logic (DL) and the relational data model (RDM). It took place as a part of the complex problem of data integration, the analysis of which can be found in [7]. The mentioned series provides an overview of the current existing approaches to address the problem of mapping. According to the result of analysis a taxonomy of research on the subject was created. This result also revealed a number of disadvantages of the existing approaches to establish one-to-one correspondences between the description logic and the relational data model. Based on this **a binary relational data model ( $RM^2$ )** was proposed as an integrating model for the creation of mappings. Complete and detailed description of it can be found in [1]. Information about the interaction between  $RM^2$  and the classical relational data model can be found in [2].

The mechanism for mapping the ALC description logic and its main extensions to  $RM^2$  was developed and described in [6], as well as classical RDM to  $RM^2$ . In the publication [6] you can get acquainted with the argumentation of the following statement: description logic can be considered as an independent data model. It also describes in detail why DL ALC is used in the developed approach, justifies the choice of ALC extensions

and outlines the way they are mapped in  $RM^2$ .

Until now a significant drawback of this approach has been the lack of any practical testing of the proposed results. The description of mappings using  $RM^2$  is purely theoretical. A real practical check can make significant changes both in the structure of the approach itself and in its main individual components, e.g. to complement or restrict the operations of a binary relational algebra ( $RA^2$ ), which is a constituent part of  $RM^2$ . In the current paper a method for checking mappings between the description logic and the binary relational data model using RDF graphs is proposed.

Section 1 is dedicated to the analysis of the number of practical works on the implementation of mappings using RDF. Section 2 formulates the problem of practical approbation of the approach to the description of mappings between DL and RDM. Section 3 outlines a method for mapping the description logic ALC and its extensions to RDF using OWL 2. Conclusions can be found in the section 4.

### Related work

The publications [1, 6] provide an overview of the current approaches to address the problem of mapping. According to the result of analysis a taxonomy of research on the subject was created. It is shown in Fig. 1.

The results given in [1, 6] will not be duplicated in the current paper. On the contrary, this section is dedicated to works that were not included in the above survey. The publications that will be overviewed in this part of the article are intended to complement the existing taxonomy. They will be classified and will take their place in the hierarchy of the body of research on the establishment of correspondences between ontologies and relational databases. This overview focuses on those methods that concern the use of RDF at RDB-to-ontology mapping or ontology-to-RDB mapping. There is a need to allocate a place in the corresponding column of the taxonomy for the researches that will be considered in the current section. Also, it is necessary to formulate a number of intermediate conclusions that are necessary for setting the task of a current work.

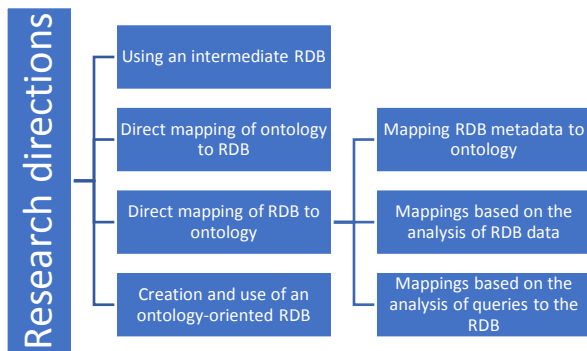


Fig. 1. The approaches to describe mappings between DL and RDB research taxonomy

The first step is to consider the fundamental work [8]. It belongs to the following section of the classification: *direct mapping of RDB in ontology*, the direction is *mappings taking into account the analysis of RDB data*. It should be noted the following features of this study:

1. A clear distinction between the concepts of "mapping RDB into ontology" and "transformation of RDB into ontology". If in the first case interaction between the existing ontology and the database is supposed, then in the other case it is supposed to create a completely new ontology based on the data and the RDM structure. The authors of the approach attribute it to the second case.

2. *Using RDF in the transformation rules*. A large amount of literature provides a set of rules that map RDB constructs directly into the ontology construct, without using an intermediate model [8]. Such an ontology is built as a result of a direct transformation from a semantically less developed database. It suffers from a number of serious shortcomings, which are mentioned by the authors of study. The difference of this approach from the others is that an independent attempt to transform the RDB structure and data into an OWL structure that is built with the help of RDF and RDFS using XSD is made. The database data is converted into regular RDF triples.

In fact, this is one of the first serious studies in the field of establishing interoperability between ontologies and RDB using RDF. The authors of the methodology do not use the R2R ML language, which has the official W3C specification. However, in the section dedicated to the source analysis the existence of various approaches that use both RDF-OWL constructs and those using R2R ML is mentioned. The R2R ML itself will be discussed below.

This approach has a serious drawback: there is no mention of how the operations of relational algebra are mapped. In their continued research [9], the authors of the technique tried to eliminate it. Based on the rules for transforming the structural part of the database and its data, the rules for mapping the relational algebra operators (expressed in SQL) into equivalent SPARQL queries were made. If we consider this approach in a complex of two works [8, 9], then it can be retrained and referred to the taxonomy section as the *direct mapping of RDB in ontology*, the direction is *mappings, taking into account the analysis of queries to the RDB*.

Generally, this approach does not stand out from the general mass of other methods in its category. It has the same drawbacks: the one-sidedness of the mechanism (although the authors note that the description of the mapping rules in the opposite direction remains in the field of future research), the absence of a formal approach, the separation of the structural and manipula-

tive parts of the RDB within a unified approach, as well as the silence that ontologies and RDB obeys two different open and closed world hypotheses. It remains unclear how the interaction between these two very different concepts will be carried out.

It should be mentioned that in studies dedicated to the mappings subject matter, a number of works that present their specification of mapping languages have appeared. For example, in [10, 11] the RDB2OWL Mapping Language specification is declared, where the specification and tools for mapping implementation are presented. However, despite the deep study of the topic and the presence of serious practical testing, RDB2OWL ML operates exclusively with the structural part of RDM. The approach pays attention to the aspect of integrity constraints but does not address the relational algebra operations converting.

Analyzing the works dedicated to the practical implementation of mappings, there was revealed the following tendency: many successful techniques increasingly prefer to work not directly with OWL, but using RDF and RDFS, in conjunction with XML data types. There are the following reasons for this. Over the years the W3C has released two official specifications for mapping mechanisms between OWL, RDF, and RDB. One of them [12] describes the mapping mechanism of the OWL ontology into an RDF triplet. The document provides an exhaustive list of rules for transforming each OWL construct into a set of RDF triplets. It also contains a description of the mechanism for transforming RDF triplets into the corresponding OWL constructs, with the necessary list of transformation rules. Another document [13] declares the R2R ML (RDB to RDF Mapping Language) language specification, which describes the mechanism for mapping a relational database to the set of RDF triplets. Thus, RDF is established by default as the intermediate stage in mappings between DL and RDM.

This is argued as follows.

1. Looking at the semantic web pie's stack architecture, it is seen that OWL and RDF are adjacent layers. That is, a very im-

portant task is to show the way of interaction between two parts of the same concept.

2. Accessing data from the “deep web”. This term refers to data that is very difficult to index with standard search engines. These include, for example, unstructured documents (pictures, scanned copies), semi-structured (CSV, PDF files), structured data sources (RDB, XML databases, NoSQL databases, LDAP directories). However, to ensure the sustainability of the applications that were developed along with the data they exploit, and to leverage the properties engineered into RDBs for decades (scalability, ACID properties, security, performance optimization), the data should remain hosted and delivered by the legacy RDBs. This situation creates a need for RDB-to-RDF methods that can access relational data and convert it into RDF triplets.

3. Linked data. Linking open data to other related pieces of data increases its value. Driven by recommendations proposed by Tim Berners-Lee [14] the Linking Open Data community project aims at extending today's web by publishing various open data sets in the RDF model and setting RDF links between data sources. This is done in order to enable developers of new programs and applications to use existing data in a new capacity, i.e. create added value by repurposing datasets, using the data in some new way, possibly beyond what data providers may have initially expected.

4. Integration of heterogeneous sources. This point has been discussed several times earlier. However, it is worth reminding the main aspects. In the modern web, there is an acute problem of using not just data, but also their semantics. Relational schemas usually convey no or poor semantics. This means that it is necessary to define somehow the semantics of the data stored in RDB in an explicit machine-readable form. Using RDF as a format for representing relational data appears as a powerful and promising method to achieve such data integration, in which RDB-to-RDF methods will play a key role.

The last three points were taken from a fundamental review [15] of methods and tools for converting RDB to RDF. The paper

describes similar studies that were conducted earlier. It notes that no RDB-to-RDF research has been conducted since the publication of R2R ML in 2012. As a result, none of the articles cited in the overview didn't review the R2R ML compliant tools.

The authors in [15] proposed the following classification of RDB-to-RDF research areas:

- description of mappings (mapping type, expression);
- implementation of mappings (when and how data is converted to RDF);
- data retrieval method (query-based methods, related data).

According to this classification, 17 approaches are ordered in the review. There is also a separate detailed analysis of the R2R ML language. The conclusions to the work indicate that it is a promising language, which, however, may not be applicable to the entire wide range of RDB-to-RDF mapping needs, leaving room for future research.

In this regard, in current paper for a practical test of the implementation of mappings between DL and RDM, the task was formulated: to describe a way to check mappings between the description logic and a binary relational data model using RDF graphs. A detailed description of the problem statement is presented in the next section.

### Problem statement

Before setting the task, it is necessary to give a brief description of the proposed theoretical approach for mappings creation between description logic and the relational data model. The results of the description of mappings DL to  $RM^2$  can be found in [1]. Here is a summary of their essence.

The mappings DL to  $RM^2$  can be split into the following components:

1. *To build a conceptual information model of DL and to transform it into  $RM^2$ .* One of the main tasks of the conceptual information model of any subject area is to define the basic concepts and to describe their properties and relationships. The ER language is one of the most used for this purpose. It assumes that a conceptual information struc-

ture is described using concepts such as entity, attribute and relationship. The conceptual information model of description logic with a detailed description of its components can be found in [6]. It also contains the  $RM^2$  scheme, which corresponds to the given ER-model.

2. *To map DL ALC into  $RM^2$ .* Any description logic consists of two conceptual parts: syntax and semantics. The latter is specified through the interpretation concept. Interpretation is a pair  $I = (\Delta, \bullet^I)$ , where  $\Delta$  is a non-empty set called the domain of interpretation and  $\bullet^I$  is an interpretation function, which assigns to each atomic concept  $A$  a set  $A^I \subseteq \Delta$  and to each atomic role  $R$  a binary relation  $R^I \subseteq \Delta \times \Delta$ . In turn, RDM operates with the set-theoretical concepts of intension and extension. The establishment of such correspondences between the components of the DL syntax and the  $RM^2$  intension, in which the semantics of the DL expression will be equal to the extension of the corresponding  $RM^2$  expression will be called *the mapping*. The theoretical representation of formulas for converting the DL ALC syntax into  $RM^2$  can be found in [6].

3. *To map DL ALC extensions into  $RM^2$ .* There are many different DL dialects. They represent a basis of ALC logic extended with one or more operations. For example, DL SHOIQ denotes the presence of all ALC syntax operators, and also includes operations of number restrictions, nominals, and there is also a role hierarchy, transitive and inverse roles. More information about how to create an ALC extensions by adding a new operation to it can be found here [16]. Also, a detailed description of which extensions were covered by the theoretical research with their mapping in  $RM^2$  can be found in [3–5].

Based on the analysis in the previous section, the following idea arose to test the mapping mechanism between DL and RDM. It is known that the mathematical basis of any ontology describing language is description logic. Thus, all constructors of concepts and roles that are present in the foundational DL are reflected in the toolbox of the corresponding language. OWL 2 is no exception. It also has the official W3C specification. Based on

this fact to set the approbation problem of the theoretical part of the description of mappings between DL and RDM the following idea is proposed. Description logic statements expressed in OWL 2 are mapped to RDF triplets using OWL-to-RDF conversion rules on the one hand, and RDB expressions to RDF triplets are mapped using R2R ML on the other hand. The resulting graphs are compared by equivalence criteria.

The idea of such an implementation is schematically shown in Fig. 2.

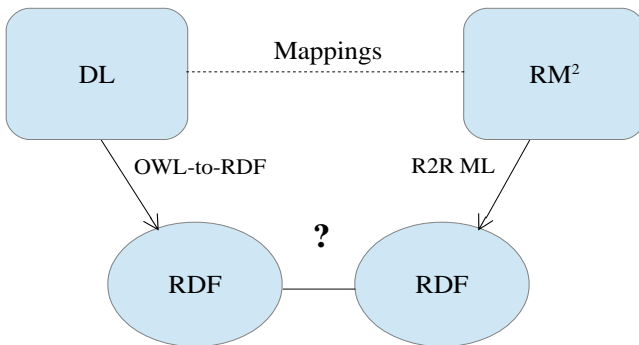


Fig. 2. The mapping method between DL and  $RM^2$  checking scheme

The idea of proof is not new. The global meaning is to transform a statement in a new theory into statements of some other existing theory. The next step is to prove the statement obtained as a result of the transformation within the framework of those methods and proofs of the established concept. If in the existing theory this statement is true, then in the area being proved the correlated expression is also true.

In our case, there is some statement of description logic that is mapped into a statement of a binary relational data model. Such a statement is represented in an OWL 2-expression from the DL side. Further such an expression is transformed into RDF triplets, forming an RDF graph, using the official W3C rules for mapping OWL-RDF. The statement is formulated in terms of RDB from the  $RM^2$  side. Such a statement is then transformed into RDF triplets using the R2R ML language [13]. The resulting triples constitute an RDF graph. Thus, as a result of such transformations, two RDF graphs are obtained. They are proposed to be compared.

If they are equivalent, then the DL-to- $RM^2$  mapping formula is true.

Here some points should be mentioned. As known [17], OWL 2 is based on the SROIQ description logic. Thus, in the documentation on OWL-to-RDF mapping [12], all issues related to both the basic syntax of DL ALC and the main extensions (concepts and roles hierarchy, nominals, number restrictions, inverse roles, DL axiomatics, as well as some of the roles restrictions) are worked out in detail. However, the scope of mapping OWL 2 to RDF is limited only by those operations that are present in DL SROIQ. The issue of mapping some of the role constructors in RDF, for which the theoretical part of DL-to- $RM^2$  mappings has been worked out, remains open.

The question of converting RDM to RDF is not so simple. Obviously, R2R ML allows you to transform the RDB structure and integrity constraints into RDF triplets. However, the way how to map the manipulative part of RDM without using the SPARQL query language have not yet been found. A mapping method of the operations of relational algebra is currently being investigated.

The key question of the approbation problem is to prove the equivalence of the resulting graphs as a result of pairwise mapping of statements DL and  $RM^2$ . In the course of research, the conclusion that an RDF graph is a special case of an ordinary graph was formulated. This means that the question of equivalence is led to proving their isomorphism. It was found that such a problem has already been investigated in [17]. It analyzes an RDF graph as a special case. Also all isomorphism criteria for the general case were studied. On the basis of these criteria three necessary and sufficient conditions for the equivalence of RDF graphs are formulated. Let's list them:

1. *Equal number of vertices.* Both graphs must contain the same number of vertices. Otherwise they are not isomorphic.
2. *Equivalence of vertices.* In a pairwise comparison, each vertex of one graph must have an equivalent in the other graph. Otherwise, such graphs are not isomorphic.

3. *Equivalence edges.* In a pairwise comparison, each edge of one graph must have an equivalent in the other graph. Otherwise, such graphs are not isomorphic.

The last question that needs to be worked out within the task is reducing the graph to a self-isomorphic. As a result of mappings, at the RDF level, a situation may arise when the vertex of one graph will semantically correspond to a subgraph from the graph with which the comparison is made. On the RDF level as a result of mappings may arise such a situation: the vertex of one graph will semantically correspond to a subgraph from the graph with which the comparison is made. Such a subgraph can consist of several vertices connected by edges. This situation should be assumed as a result of such a fact: when a statement is mapped to RDF, a large number of anonymous (empty) nodes arise, which, nevertheless, have their own semantic purpose. Thus, the question of reducing an RDF graph to a self-isomorphic remains open.

### Mapping DL to RDF

It is known [18] that all modern description logics are based on the simplest version of DL ALC. This means that it is fully included in the DL SROIQ. Therefore, OWL 2 uses all the functionality of ALC.

The syntax for this logic is defined as follows:

$$\top \mid \perp \mid A \mid \neg C \mid C \sqcap D \mid C \sqcup D \mid \exists R.C \mid \forall R.C,$$

where A – atomic concept, R – atomic role, C, D – concept.

The concept of DL semantics does not play a significant role in the context of mapping to RDF, so in this article there is no focus on this point.

The OWL 2 ontology and the mapped graph are related as follows:

$$G = T(O),$$

where G – graph, O – ontology, T – the mapping function.

Before proceeding to the description of the mappings, a number of designations

should be given. To describe OWL 2 constructs the OWL Abstract Syntax will be used. To describe RDF expressions, the standard triples and serialization to N3 notation will be used.

Table 1 shows the notation for the main namespaces.

The notation T (SEQ  $y_1, \dots, y_n$ ) shows the translation of a sequence of the OWL objects from a structural specification into an RDF collection. A few words should be said about this way of organizing resources. An anonymous node, which belongs to the rdf: list class, and two types of predicates act as a subject to create a collection. The predicates are as follows:

- rdf: first – the first element of the collection (head);
- rdf: rest – the link to sub-collection (tail).

Table 1. Namespace notation

Prefix name	Expansion
@prefix rdf:	http://www.w3.org/1999/02/22-rdf-syntax-ns#
@prefix rdfs:	http://www.w3.org/2000/01/rdf-schema#
@prefix owl:	http://www.w3.org/2002/07/owl#
@prefix xsd:	http://www.w3.org/2001/XMLSchema#

The head of the collection points to its first element (an RDF triple object). The tail contains the remaining items, which are also organized into a collection (a sub-collection of the main collection). It looks the same way – a node (subject) and two predicates (head and tail), one of which points to the object, and the second points to the remaining elements. The tail that contains the last element of the collection points to the built-in resource rdf: nil.

$\_$ : x will denote an anonymous RDF triplet node.

So, let's describe the mappings of the description logic ALC to RDF using OWL 2. Table 2 shows the mapping rules for all components of the ALC syntax.

Table 2. DL ALC to RDF mapping rules

ALC	OWL 2	RDF
$\top$	owl:Thing	-
$\perp$	owl:Nothing	-
C, D	Declaration (Class (C)), Declaration (Class (D))	T(C) rdf:type owl:Class, T(D) rdf:type owl:Class
R	Declaration (ObjectProperty(R)) Declaration (DatatypeProperty(R))	T(R) rdf:type owl:ObjectProperty, T(R) rdf:type owl:DatatypeProperty
$\neg C$	ObjectComplementOf(C)	$\_x$ rdf:type owl:Class $\_x$ owl:complementOf T(C)
$C \sqcap D$	ObjectIntersectionOf(C, D)	$\_x$ rdf:type owl:Class $\_x$ owl:intersectionOf T(SEQ C, D)
$C \sqcup D$	ObjectUnionOf(C, D)	$\_x$ rdf:type owl:Class $\_x$ owl:unionOf T(SEQ C, D)
$\exists R.C$	ObjectSomeValuesFrom(R C)	$\_x$ rdf:type owl:Restriction $\_x$ owl:onProperty T(R) $\_x$ : owl:someValuesFrom T(C)
$\forall R.C$	ObjectAllValuesFrom(R C)	$\_x$ rdf:type owl:Restriction $\_x$ owl:onProperty T(R) $\_x$ : owl:allValuesFrom T(C)

The concepts  $\top$  and  $\perp$  are represented in OWL by the special classes – owl:Thing and owl:Nothing. When constructing RDF

triplets, these classes are used in the same way as within OWL itself.

The images of RDF triplets and their serialization to N3 notation are shown below.

### Definition of concepts C and D

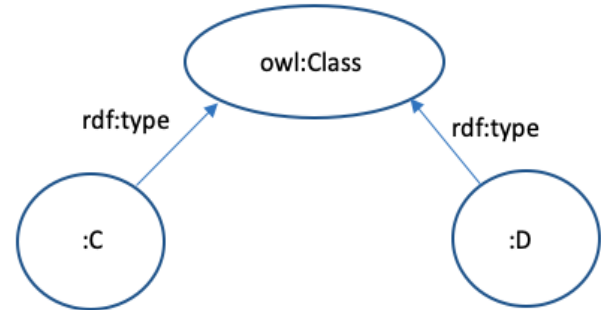


Fig. 3. RDF triplets of C and D definition image

@prefix : <http://example.com/Ch.owl#>  
:C rdf:type owl:Class.  
:D rdf:type owl:Class.

### Definition of the role R



Fig. 4. RDF triplets of the object property R definition image



Fig. 5. RDF triplets of the datatype property R definition image

Since the role R can describe both an object property and a datatype property, two RDF triplets for this element were defined. N3 serialization format for two triples is also present.

@prefix : <http://example.com/Ch.owl#>

:R rdf:type owl:ObjectProperty.  
 @prefix : <http://example.com/Ch.owl#>  
 :R rdf:type owl:DatatypeProperty.

**Complement of concept C**

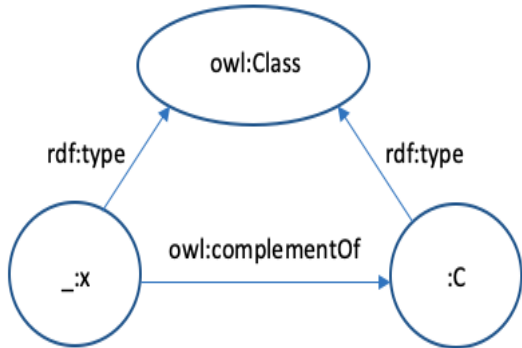


Fig. 6. RDF triplets of the complement of concept C image

@prefix : <http://example.com/Ch.owl#>  
 \_:x rdf:type owl:Class;  
 owl:complementOf :C.

**Concept intersection  $C \sqcap D$**

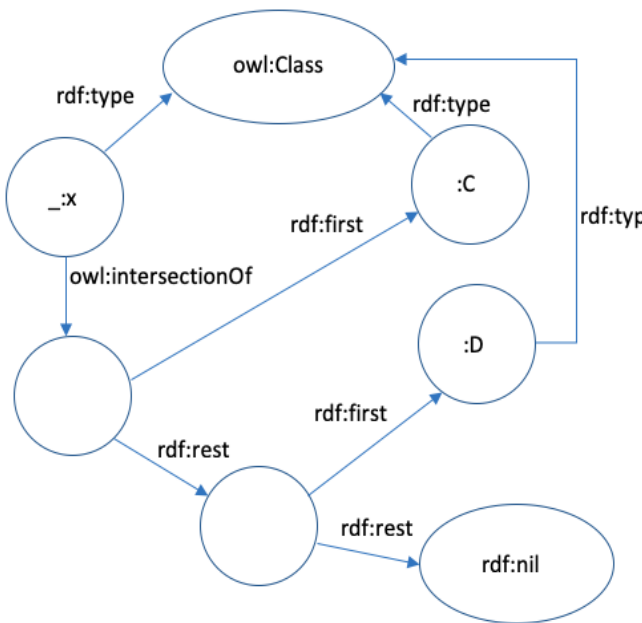


Fig. 7. RDF triplets of the concept intersection image

@prefix : <http://example.com/Ch.owl#>  
 \_:x rdf:type owl:Class;  
 \_:x owl:intersectionOf: (:C :D).

**Concept union  $C \sqcup D$**

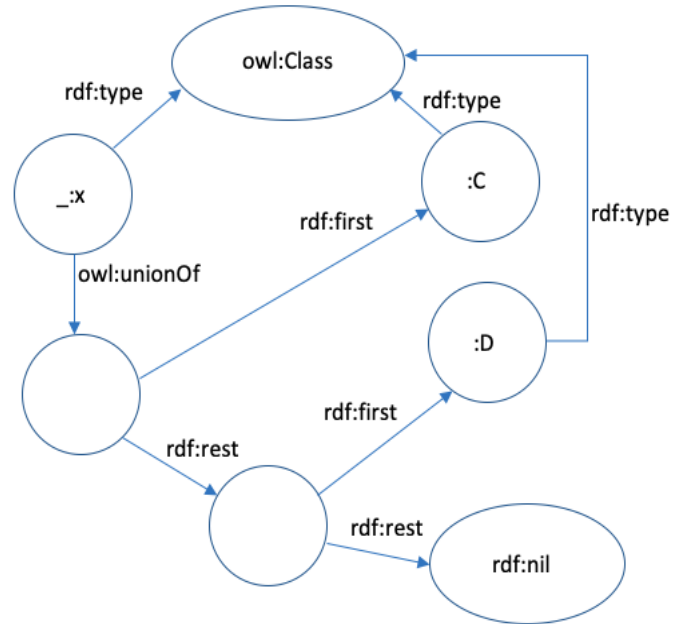


Fig. 8. RDF triplets of the concept union image

@prefix : <http://example.com/Ch.owl#>  
 \_:x rdf:type owl:Class;  
 \_:x owl:unionOf: (:C :D).

**Existential quantification  $\exists R.C$**

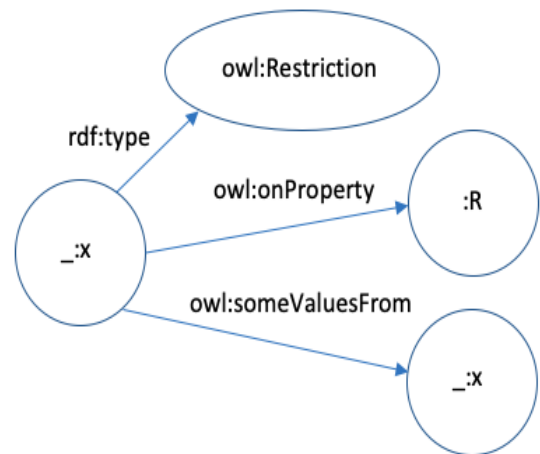


Fig. 9. RDF triplets of the existential quantification image

@prefix : <http://example.com/Ch.owl#>  
 \_:x rdf:type owl:Restriction;  
 owl:onProperty :R;  
 owl:someValuesFrom:C.



**Value restriction  $\forall R.C$**

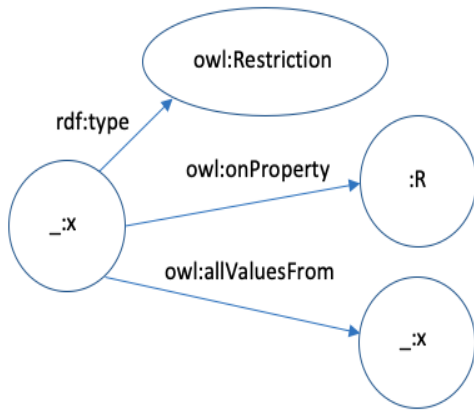


Fig. 10. RDF triplets of the value restriction image

```
@prefix : http://example.com/Ch.owl#
_:x rdf:type owl:Restriction;
      owl:onProperty :R;
      owl:allValuesFrom :C.
```

In cases of mapping the existential quantification and value restriction concepts, where the role R reflects a datatype property, T (DR) instead of the node :C is used, where DR is the data range.

**Number restrictions, nominals**

The following constructors are called number restrictions. If R is a role, C is a concept, and  $n \geq 0$  is a natural number, then:

- $(\leq nR)$  и  $(\geq nR)$  – at-least and at-least number restrictions;
- $(\leq nR.C)$  и  $(\geq nR.C)$  – qualified number restrictions.

In the OWL there is an owl:cardinality constraint [19]. It describes a class of all individuals that have exactly N semantically distinct values (individuals or data values) for the property concerned, where N is the value of the cardinality constraint. This construct is in fact redundant as it can always be replaced by a pair of matching owl:minCardinality and owl:maxCardinality constraints with the same value. It is included as a convenient shorthand for the user. Table 3 shows the mapping rules for number restrictions and nominal.

Table 3. Number restrictions and nominal to RDF mapping rules

Extensions	OWL 2	RDF
1	2	3
$=nR$	ObjectExactCardinality(n R)	_:x rdf:type owl:Restriction. _:x owl:onProperty T(R). _:x owl:cardinality «n»^^xsd:nonNegativeInteger.
$=nR.C$	ObjectExactCardinality(n R C)	_:x rdf:type owl:Restriction. _:x owl:onProperty T(R). _:x owl:cardinality «n»^^xsd:nonNegativeInteger. _:x owl:onClass T(C)
$\leq nR$	ObjectMinCardinality(n R)	_:x rdf:type owl:Restriction. _:x owl:onProperty T(R). _:x owl:minCardinality «n»^^xsd:nonNegativeInteger.
$\leq nR.C$	ObjectMinCardinality(n R C)	_:x rdf:type owl:Restriction. _:x owl:onProperty T(R). _:x owl:minQualifiedCardinality «n»^^xsd:nonNegativeInteger. _:x owl:onClass T(C)
$\geq nR$	ObjectMaxCardinality(n R)	_:x rdf:type owl:Restriction. _:x owl:onProperty T(R). _:x owl:maxCardinality «n»^^xsd:nonNegativeInteger.

1	2	3
$\geq nR.C$	Object- MaxCardinality( $n R C$ )	<code>_:x rdf:type owl:Restriction.</code> <code>_:x owl:onProperty T(R).</code> <code>_:x owl:maxQualifiedCardinality «n»^^xsd:nonNegativeInteger.</code> <code>_:x owl:onClass T(C)</code>
$\{a\}$	ObjectOneOf( $a$ )	<code>_:x rdf:type owl:Class.</code> <code>_:x owl:oneOf T(SEQ a).</code>

**Number restriction =nR**

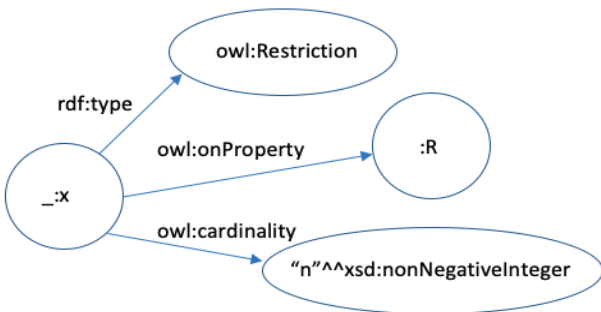


Fig. 11. RDF triplet of the number restriction =nR

@prefix : <http://example.com/Ch.owl#>  
 \_:x rdf:type owl:Restriction;  
 owl:onProperty :R;  
 owl:cardinality «n»^^xsd:nonNegativeInteger.

**Number restriction =nR.C**

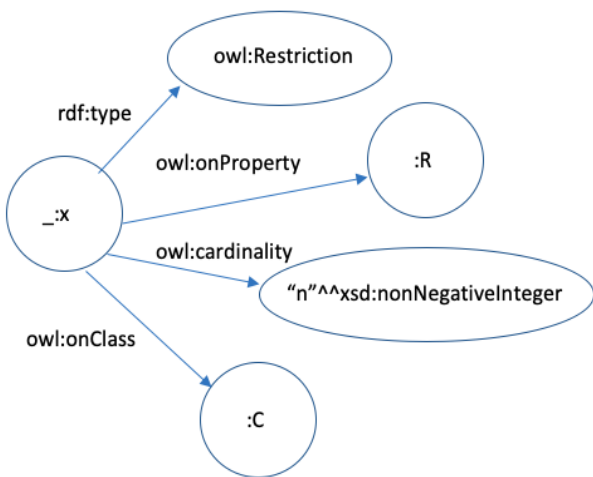


Fig. 12. RDF triplet of the number restriction =nR.C

@prefix : <http://example.com/Ch.owl#>  
 \_:x rdf:type owl:Restriction;  
 owl:onProperty :R;  
 owl:cardinality «n»^^xsd:nonNegativeInteger.  
 owl:onClass :C.

**At-least number restriction ≤nR**

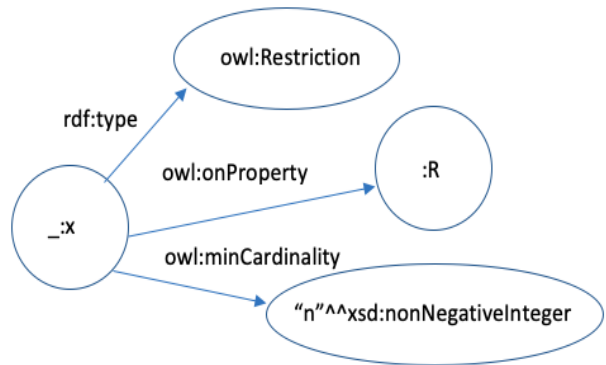


Fig. 13. RDF triplet of the at-least number restriction

@prefix : <http://example.com/Ch.owl#>  
 \_:x rdf:type owl:Restriction;  
 owl:onProperty :R;  
 owl:minCardinality «n»^^xsd:nonNegativeInteger.

**Qualified number restriction ≤nR.C**

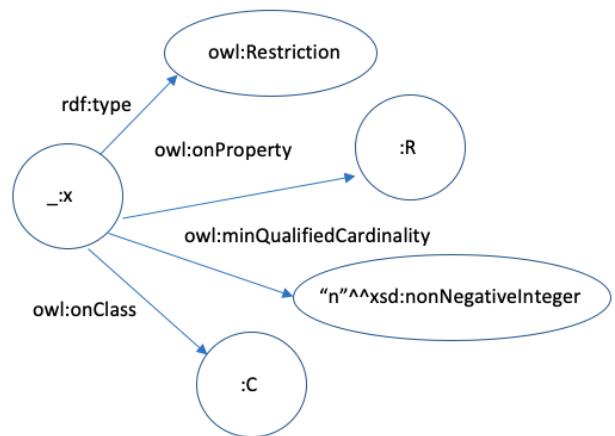


Fig. 14. RDF triplet of the qualified number restriction ≤nR.C

@prefix : <http://example.com/Ch.owl#>  
 \_:x rdf:type owl:Restriction;  
 owl:onProperty :R;

owl:minQualifiedCardinality  
 «n»^^xsd:nonNegativeInteger.  
 owl:onClass :C.

**At-last number restriction  $\geq nR$**

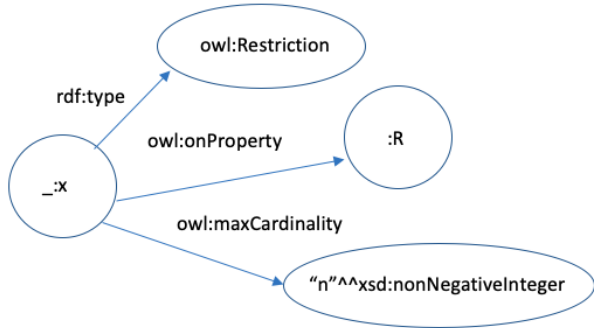


Fig. 15. RDF triplet of the at-last number restriction

@prefix : <http://example.com/Ch.owl#>  
 \_:x rdf:type owl:Restriction;  
 owl:onProperty :R;  
 owl:maxCardinality «n»  
 xsd:nonNegativeInteger.

**Qualified number restriction  $\geq nR.C$**

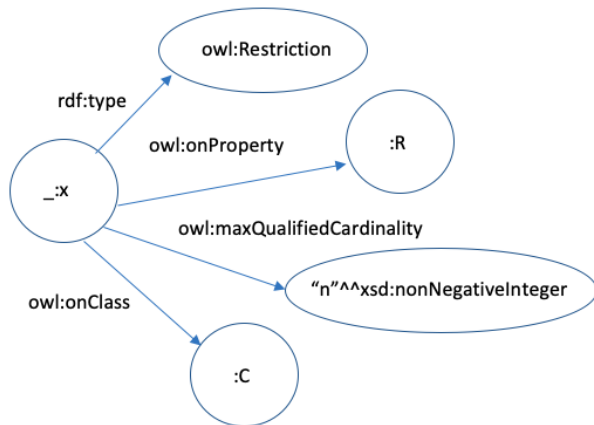


Fig. 16. RDF triplet of the qualified number restriction  $\geq nR.C$

@prefix : <http://example.com/Ch.owl#>  
 \_:x rdf:type owl:Restriction;  
 owl:onProperty :R;  
 owl:maxQualifiedCardinality «n»  
 xsd:nonNegativeInteger.  
 owl:onClass :C.

**Nominal {a}**

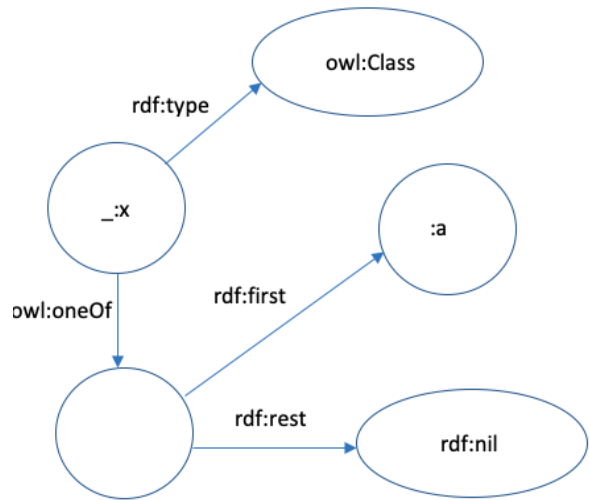


Fig. 17. RDF triplet of the nominal

@prefix : <http://example.com/Ch.owl#>  
 \_:x rdf:type owl:Class;  
 owl:oneOf(:a).

**Role constructors**

If R and S are roles, and C is a concept, then the following expressions are also roles:  $R^-$  (inverse role),  $\neg R$  (complement),  $R \sqcap S$  (intersection),  $R \sqcup S$  (union),  $R \circ S$  (composition),  $R^+$  (transitive closure),  $R^*$  (reflexive-transitive closure),  $id(C)$  (role identity).

In OWL 2 through all the role constructors only inverse role is present. This means that the mapping rules exist only for this operation. It looks like this in the Table 4.

Table 4. Inverse role to RDF mapping rule

Constructors	OWL 2	RDF
$R^-$	InverseObjectProperties ( $R^-$ R)	$T(R^-)$ owl:inverseOf T(R)



Fig. 18. RDF triplet for inverse role

@prefix : <http://example.com/Ch.owl#>

:R- owl:inverseOf :R

The issue of mapping the remaining role operators to RDF remains open.

**DL axiomatics**

The DL axioms include the following rules:

- concept nesting  $C \sqsubseteq D$ ;
- concept equivalence  $C \equiv D$ ;
- role nesting  $R \sqsubseteq S$ ;
- role equivalence  $R \equiv S$ ;
- concept individual equivalence  $a = b$ .

Mapping rules are represented in table 5.

Table 5. DL axiomatics to RDF mapping rules

Axiom	OWL 2	RDF
$C \equiv D$	Equivalent-Classes (C D)	T(R) owl:equivalentClass T(D)
$C \sqsubseteq D$	SubClassOf(C D)	T(R) rdfs:subClassOf T(D)
$R \sqsubseteq S$	SubProjectPropertyOf(R S) SubDataPropertyOf(R S)	T(R) rdfs:subPropertyOf T(S) T(R) rdfs:subPropertyOf T(S)
$R \equiv S$	Equivalent-ObjectProperties(R S) Equivalent-DataProperties(R S)	T(R) owl:equivalentProperty T(S) T(R) owl:equivalentProperty T(S)
$a = b$	SameIndividual (a b)	T(a) owl:sameAs T(b)

**Concept equivalence**



Fig. 19. RDF triplet for concept equivalence

@prefix : <http://example.com/Ch.owl#>

:C owl:equivalentClass :D.

**Concept nesting**

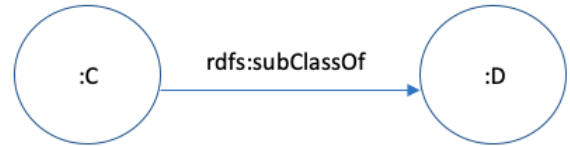


Fig. 20. RDF triplet for concept nesting

@prefix : <http://example.com/Ch.owl#>

:C rdfs:subClassOf :D.

**Role equivalence**



Fig. 21. RDF triplet for role equivalence

@prefix : <http://example.com/Ch.owl#>

:R owl:equivalentProperty :S

**Role nesting**



Fig. 22. RDF triplet for role nesting

@prefix : <http://example.com/Ch.owl#>

:R rdfs:subPropertyOf :S

**Concept individual equivalence**

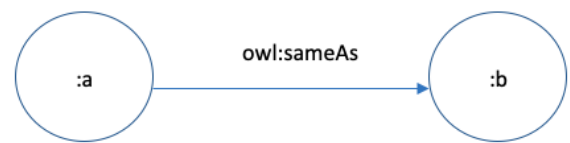


Fig. 23. RDF triplet for concept individual equivalence

@prefix : <http://example.com/Ch.owl#>

:a owl:sameAs :b

**Conclusion**

A method for checking mappings of description logic to the binary relational data model using transformations to RDF is described in the article. The description of the approach is given. Bottlenecks and potential

problems are identified. Mappings for the DL axiomatics, as well as for all those constructors of concepts and roles that are implemented in the OWL 2, based on the W3C OWL 2-to-RDF mapping rules are provided in the publication. The issue of mapping several of role constructors to RDF remains open. Mapping a binary relational data model to RDF is in the field for further research.

## References

1. Andon P., Reznichenko V., Chystiakova I. Mapping of Description Logic to the Relational Data Model. *Cybernetics and Systems Analysis*. 2017. 53 (6). P. 963–978.
2. Reznichenko V., Chystiakova I. Binary Relational Data Model. *Problems in Programming*. 2017. Vol. 2 (4). P. 96–105.
3. Chystiakova I. Integration of the description logics axiomatic into relational data model. *Problems in Programming*. 2017. Vol. 1(3). P. 51–58.
4. Chystiakova I. Integration of the description logics with extensions into relational data model. *Problems in Programming*. 2016. N 4. P. 58–65.
5. Reznichenko V., Chystiakova I. Integration of the family of extended description logics with relational data model. *Problems in Programming*. 2016. N 2–3. P. 38–47.
6. Reznichenko V., Chystiakova I. Mapping of the Description Logics ALC into the Binary Relational Data Structure. *Problems in Programming*. 2015. N 4. P. 13–30.
7. Chystiakova I. Ontology-oriented data integration on the Semantic Web. *Problems in Programming*. 2014. N 2–3. P. 188–196.
8. Hazber M.A.G., LI R., GU X., XU G. Integration Mapping Rules: Transforming Relational Database to Semantic Web Ontology. *Applied Mathematics & Information Sciences*. 2016. Vol. 3(10). P. 881–901.
9. Hazber M., LI B., XU G., Mosleh M., GU X., LI Y. An Approach for Generation of SPARQL Query from SQL Algebra based Transformation Rules of RDB to Ontology. *Journal of Software*. San Bernardino. 2018. CA. USA. 2018. Vol. 13(11). P. 573–599.
10. Cerans K., Bumans G. RDB2OWL: a RDB-to-RDF/OWL Mapping specification Language. Proceedings of the 2011 conference on Databases and Information Systems VI: Selected Papers from the Ninth International Baltic Conference, DB&IS 2010. Riga, Latvia, 5–7 July 2010. P. 139–152.
11. Cerans K., Bumans G. RDB2OWL: A language and tool for database to ontology mapping. Proceedings of the 27th International Conference on Advanced Information Systems Engineering (CAiSE 2015). Stockholm, Sweden, 8–12 June 2015. Vol. 1367. P. 81–88.
12. OWL 2 Web Ontology Language. Mapping to RDF Graphs (Second Edition). [Online] December 2012. Available from: [https://www.w3.org/TR/owl2-mapping-to-rdf/#Translation\\_of\\_Axioms\\_without\\_Annotations](https://www.w3.org/TR/owl2-mapping-to-rdf/#Translation_of_Axioms_without_Annotations). [Accessed: 20 February 2020].
13. R2RML: RDB to RDF Mapping Language. [Online] September 2012. Available from: <https://www.w3.org/TR/r2rml/>. [Accessed: 20 February 2020].
14. Berners-Lee T. Linked data, in design issues of the WWW. [Online] 2006. Available from: <https://www.w3.org/DesignIssues/LinkedData.html>. [Accessed: 20 February 2020].
15. Michel F., Montagnat J., Zucker C.F. A survey of RDB to RDF translation approaches and tools. [Research Report] I3S. 2014. [Online]. 2014. Available from: [https://hal.archives-ouvertes.fr/hal-00903568/file/Rapport\\_Rech\\_I3S\\_v2\\_-\\_Michel\\_et\\_al\\_2013\\_-\\_A\\_survey\\_of\\_RDB\\_to\\_RDF\\_translation\\_approaches\\_and\\_tools.pdf](https://hal.archives-ouvertes.fr/hal-00903568/file/Rapport_Rech_I3S_v2_-_Michel_et_al_2013_-_A_survey_of_RDB_to_RDF_translation_approaches_and_tools.pdf). [Accessed 20 February 2020].
16. Description logics with axiomatics. [Online] 2017-2018. Available from: [http://lpcs.math.msu.su/~zolin/dl/pdf/DL\\_07\\_SHIQ.pdf](http://lpcs.math.msu.su/~zolin/dl/pdf/DL_07_SHIQ.pdf)
17. Kontchakov R., Zakharyshev M. An introduction to description logics and query rewriting. Reasoning Web International Summer School. Birmingham, UK. 8 September 2014. Vol. 8714. P. 195–244.
18. Baader F. et al. The Description Logic Handbook. 2003. P. 51–55.
19. OWL Web Ontology Language Reference [Online] 2004. Available from: <https://www.w3.org/TR/owl-ref/>

## Література

1. Andon P., Reznichenko V., Chystiakova I. Mapping of Description Logic to the Relational Data Model. *Cybernetics and Systems Analysis*. 2017. 53 (6). P. 963–978.
2. Reznichenko V., Chystiakova I. Binary Relational Data Model. *Problems in Programming*. 2017. Vol. 2 (4). P. 96–105.

3. Chystiakova I. Integration of the description logics axiomatic into relational data model. *Problems in Programming*. 2017. Vol. 1(3). P. 51–58.
4. Chystiakova I. Integration of the description logics with extensions into relational data model. *Problems in Programming*. 2016. № 4. P. 58–65.
5. Reznichenko V., Chystiakova I. Integration of the family of extended description logics with relational data model. *Problems in Programming*. 2016. № 2–3. P. 38–47.
6. Reznichenko V., Chystiakova I. Mapping of the Description Logics ALC into the Binary Relational Data Structure. *Problems in Programming*. 2015. № 4. P. 13–30.
7. Chystiakova I. Ontology-oriented data integration on the Semantic Web. *Problems in Programming*. 2014. № 2–3. P. 188–196.
8. Hazber M.A.G., LI R., GU X., XU G. Integration Mapping Rules: Transforming Relational Database to Semantic Web Ontology. *Applied Mathematics & Information Sciences*. 2016. Vol. 3(10). P. 881–901.
9. Hazber M., LI B., XU G., Mosleh M., GU X., LI Y. An Approach for Generation of SPARQL Query from SQL Algebra based Transformation Rules of RDB to Ontology. *Journal of Software. San Bernardino*. 2018. CA. USA. 2018. Vol. 13(11). P. 573–599.
10. Cerans K., Bumans G. RDB2OWL: a RDB-to-RDF/OWL Mapping specification Language. Proceedings of the 2011 conference on Databases and Information Systems VI: Selected Papers from the Ninth International Baltic Conference, DB&IS 2010. Riga, Latvia, 5-7 july 2010. P. 139–152.
11. Cerans K., Bumans G. RDB2OWL: A language and tool for database to ontology mapping. Proceedings of the 27th International Conference on Advanced Information Systems Engineering (CAiSE 2015). Stockholm, Sweden, 8-12 june 2015. Vol. 1367. P. 81–88.
12. OWL 2 Web Ontology Language. Mapping to RDF Graphs (Second Edition). [Online] December 2012. Available from: [https://www.w3.org/TR/owl2-mapping-to-rdf/#Translation\\_of\\_Axioms\\_without\\_Annotations](https://www.w3.org/TR/owl2-mapping-to-rdf/#Translation_of_Axioms_without_Annotations). [Accessed: 20 february 2020].
13. R2RML: RDB to RDF Mapping Language. [Online] September 2012. Available from: <https://www.w3.org/TR/r2rml/>. [Accessed: 20 february 2020].
14. Berners-Lee T. Linked data, in design issues of the WWW. [Online] 2006. Available from: <https://www.w3.org/DesignIssues/LinkedData.html>. [Accessed: 20 february 2020].
15. Michel F., Montagnat J., Zucker C.F.. A survey of RDB to RDF translation approaches and tools. [Research Report] I3S. 2014. [Online]. 2014. Available from: [https://hal.archives-ouvertes.fr/hal-00903568/file/Rapport\\_Rech\\_I3S\\_v2\\_-\\_Michel\\_et\\_al\\_2013\\_-\\_A\\_survey\\_of\\_RDB\\_to\\_RDF\\_translation\\_approaches\\_and\\_tools.pdf](https://hal.archives-ouvertes.fr/hal-00903568/file/Rapport_Rech_I3S_v2_-_Michel_et_al_2013_-_A_survey_of_RDB_to_RDF_translation_approaches_and_tools.pdf). [Accessed 20 february 2020].
16. Description logics with axiomatics. [Online] 2017-2018. Available from: [http://lpcs.math.msu.su/~zolin/dl/pdf/DL\\_07\\_SHIQ.pdf](http://lpcs.math.msu.su/~zolin/dl/pdf/DL_07_SHIQ.pdf)
17. Kontchakov R., Zakharyshev M. An introduction to description logics and query rewriting. Reasoning Web International Summer School. Birmingham, UK. 8 september 2014. Vol. 8714. P. 195–244.
18. Baader F. et al. The Description Logic Handbook. 2003. P. 51–55.
19. OWL Web Ontology Language Reference [Online] 2004. Available from: <https://www.w3.org/TR/owl-ref/>

Received 18.10.2020

**About the author:**

*Inna Chystiakova*,  
junior researcher at the Institute of software systems of NASU.  
The number of publications in Ukrainian journals – 10.  
The number of publications in foreign journals – 1.  
Hirsh index is 5.  
<https://orcid.org/0000-0001-7946-3611>.

**Affiliation:**

Institute of software systems of NASU  
03187, Kyiv,  
pr. Glushkova, 40, build 5.  
Tel.: +38(066)8477784.  
E-mail: [inna\\_islyamova@ukr.net](mailto:inna_islyamova@ukr.net).