Representation of Knowledge patterns for Semantic Web

Rostislav Miarka[1]

[1] **Department of Informatics and Computers, Faculty of Science, University of Ostrava**
30. dubna 22, 701 03 Ostrava, Czech Republic

Abstract

This article presents representation of knowledge patterns in RDF language. The approach to knowledge representation can be used in Semantic Web as a tool of finding some additional RDF assertions in the frame of linked data. The article introduces the term "knowledge pattern" and dividing the knowledge patterns into two groups – to Top-level knowledge patterns and to Domain knowledge patterns. Examples of top-level knowledge patterns for sentences in the English language are also a part of this article.

Keywords: *Knowledge patterns, RDF graph language, Semantic web, English sentence.*

1. Introduction

The term "Semantic web" refers to the vision of Web of linked data. Nowadays, the Semantic web is also called Web of Data, which is enabled by Semantic web technologies. Among the Semantic web technologies are RDF, OWL, SKOS and SPARQL. The main goal of Linked data is to allow users to share structured data in the web environment. The term "Linked data" was introduced by Tim Berners-Lee in his talk named Linked Data Web Architecture. This term represents the way of publishing and linking of structured data on the web.

Linked data uses RDF (Resource Description Framework) in two ways. The first way is the use of RDF data model to publish structured data on the web. The second way is the use of RDF links to link data sources.

2. Knowledge patterns

Knowledge patterns were first introduced in [1] and they are closely related to the creation of ontologies or knowledge bases. The term knowledge pattern represents a general structure (pattern) of knowledge. The pattern itself is not a part of the target knowledge base or ontology. While using a knowledge pattern, the general terms from pattern are renamed to special terms from the modeled domain. Renaming of terms is called morphism. The main purpose of knowledge patterns is a reuse of knowledge.

We propose to divide knowledge patterns to two groups – Top-level knowledge patterns and Domain knowledge patterns.

2.1 Top-level knowledge patterns

The original idea of knowledge patterns was based on the fact that the pattern itself is not a part of the target knowledge base and it is integrated to the knowledge base by renaming its symbols. Knowledge patterns which correspond to this description are called Top-level knowledge patterns and they form the first group of knowledge patterns. Such knowledge patterns can be used in any knowledge base or ontology.

2.2 Domain knowledge patterns

The second group of knowledge patterns is called Domain knowledge patterns. Nowadays, many ontologies, vocabularies or knowledge bases are created by web ontological languages. Languages RDF+RDFS and OWL are the most important web ontological languages. In these languages, the ontology contains definition of classes and relation among them. Relations are also called properties. The OWL language enables using two types of properties – object properties and datatype properties. Object property defines relation between two classes; datatype property defines attributes of a class. The attribute is represented by a data type, such as string, integer or date.

The particular structure of classes and properties creates knowledge pattern on lower level of abstraction. While using this type of knowledge patterns, instances of classes (individuals) and instances of properties are created. These instances represent concrete objects from the modeled domain. This type of knowledge pattern is called Domain knowledge pattern.

We propose to represent knowledge patterns in RDF+RDFS[1] languages.

[1] In the following, RDF(S) will be used as an abbreviation for RDF+RDFS.

3. Representation of knowledge patterns in RDF(S)

The basic building block of RDF(S) is an RDF triple, which represents one statement. The statement is in form

subject – predicate – object.

The subject represents a term which is described in the statement. The predicate represents property assigned to the subject. The object represents the value of this property. It can be other term or a literal value (string, integer, date). Terms and properties in RDF(S) are identified by URI (Uniform Resource Identifier) references. RDF(S) contains only binary relations, more complex relations must be decomposed to a set of binary relations. The RDF(S) language can be represented in two forms – as a graph and in a text form.

The RDF graph language is used to create graph representation of RDF. Ontology or knowledge base in the RDF graph language is represented by a directed graph. Nodes in this graph represent subjects and objects of statements, arcs represent relations (predicates). The basic building block of an RDF graph is an RDF triple, which represent one statement. The statement is in form

subject – predicate – object.

Its graph representation is shown in next figure.

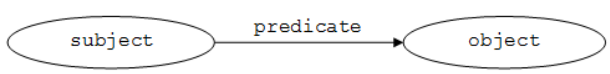

Figure 1. RDF triple.

Text form of RDF is called RDF serialization. It can have more forms. Among these forms is RDF/XML, N3 notation, N-triples, RDFa. Serialization called RDF/XML is the mostly used type of serialization. It is based on the XML language.

For representation of knowledge patterns, we propose to use the RDF(S) language [6], [7]. In the case of graph representation, it is used the RDF graph language which is extended by quantifiers and enables to state negation [4]. A classical RDF graph uses only solid lines. To distinguish the knowledge patterns from the classical RDF statements, we use dashed lines. The example of a classical RDF triple and a triple which represents knowledge pattern is shown in Figure 2.

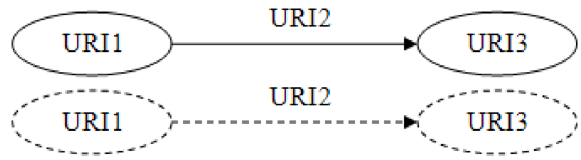

Figure 2. Classical RDF triple (above) and RDF triple representing knowledge pattern (below).

The important part of using knowledge patterns is morphism. By the help of morphism, the general terms from the pattern are mapped to special terms from the problem domain. Morphism is represented by the relation of specialization – the general term is renamed to a special term from the domain of interest. Mapping of one term will be represented by the one classical RDF triple (with a solid line). The subject of this triple will be a special term from the domain of interest, the predicate will be "isa" and the object will be the general term from the pattern. The example of morphism of one term is shown in Figure 3.

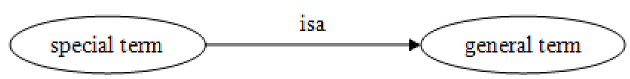

Figure 3. Morphism.

Representation of knowledge patterns in a text form will be described by the RDF/XML serialization. Vocabulary for description of knowledge patterns was defined. The schema of this vocabulary is shown in next figure.

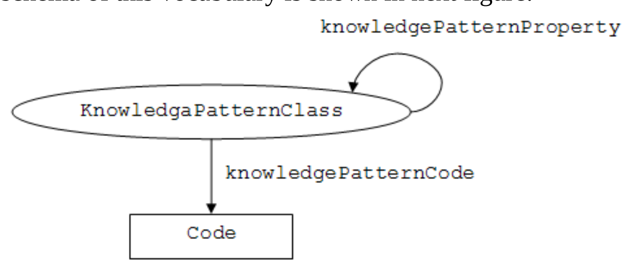

Figure 4. Schema of vocabulary for knowledge patterns description.

The class called *KnowledgePatternClass* represents one class (i.e. one term) from knowledge pattern. It is an abstract term which represents any class. All classes which belong to knowledge pattern are subclasses of *KnowledgePatternClass*. Classes in the knowledge pattern are connected by properties. An abstract property which connects two classes in the pattern is called *knowledgePatternProperty*. All properties which belong to knowledge pattern are subproperties of this property. The property *knowledgePatternCode* assigns the code of the knowledge pattern (literal value) to all classes form the pattern. The code of patterns is its identifier. While using concrete knowledge patters, morphism of terms is defined by the relation *isa*. This relation defines mapping of one special term to the general term.

The vocabulary for description knowledge patterns is available online: http://www.r-miarka.net/kp.rdf.

4. Examples of knowledge patterns

In the rest of the article, there will be presented examples of knowledge patterns, concretely of top-level knowledge patterns. There will be examples of knowledge patterns for conversion of sentences from the English language to RDF(S).

Words of a natural language form vocabulary of this language. Words are divided to word classes, such as noun, pronoun, verb etc. Grammar of a language determines the way of constructing a sentence. Each language has its own grammar. Grammar of a language does not work with work classes; it works with parts of sentences, which are also called constituents of sentences. There are two basic parts of sentence – the subject and the predicate. These two parts are to be found in almost each sentence. For example, imperative sentences ("Run!", "Stop!" etc.) belong to exceptions of this rule. Apart from subject and predicate, there are additional parts, which extend the sentence – object, attribute and adverbial complement (adverb).

For the process of construction of sentence, it is important, which part the particular word represents. The term word order is closely related to this process. Word order of language determines the order of parts in a sentence. There exist two basic types of word order – fixed word order and free word order. Fixed word order has relatively strict rules for ordering parts in sentence. This type of word order is used by Germanic languages, e.g. English. Free word order allows changing the order of parts in sentence, according to the actual context. This type of word order is typical for languages which enable declension and inflexion. Among these languages is Czech language.

The essential parts of each sentence are subject and predicate. To mark the basic types of word orders, we use three parts – subject, predicate and object. All three parts are marked with a letter. Subject is marked by the letter S, predicate is marked by the V (verb) and object is marked by O. Combination of these three basic parts determines the type of word order [2], [3]. There are six basic types of word order: SVO (subject verb object), SOV (subject object verb), VOS (verb object subject), VSO (verb subject object), OSV (object subject verb) and OVS (object verb subject). Languages using the SVO word order include English, Romance languages, Bulgarian and Chinese. Languages using the SOV word order include Japanese, Mongolian, Turkish and Korean. Word order marked as VSO is used by the following languages: Classical Arabic, Insular Celtic languages and Hawaiian. Word order VOS is used in the Fijian language (Fiji) and the Malagasy language (Madagascar). The OSV word order is used by languages Xavante (Brazil) and Warao (Venezuela). The OVS word order is used by the Hixkaryana language (Brazil).

4.1 Structure of English sentence

The English language uses a fixed word order marked as SVO (subject verb object). The letter O in this abbreviation marks a direct object. The basic word order can be a little different because in the sentence can appear an auxiliary verb (will, can, would etc.) or indirect object. Sentences can express different things. It can state a fact – declarative sentences (positive or negative). It can ask about some things – questions. It can give commands – imperative sentences. Because this article is focused on knowledge patterns, which are closely related to ontologies or knowledge bases, the only type of a sentence which is important is declarative sentence. Questions or imperative sentences are not a part of any knowledge base or ontology.

Declarative sentences can state positive facts or negative facts. The structure of both positive and negative sentence is quite similar in the English language. A sentence in a negative form contains auxiliary words "do not" in addition.

The simplest type of positive declarative sentence is a bare sentence, which contains only a subject and predicate. Examples of this type of a sentence are "It snows.", "I hope." etc.

A more complex type of sentence is that which contains an object in addition. This type of a sentence is in form SVO (subject verb object). Examples of this type of a sentence are "I know Michael.", "I like oranges.", etc. Another type of a sentence contains an adverbial complement in addition. It can be an adverbial complement of manner, place or time. A sentence can contain one of these complements or a combination of them. A sentence containing all three types of adverbial complements has its word order marked as SVOMPT (subject verb object manner place time). Examples of this type of a sentence are "I play the piano daily.", "I speak English very well." etc. Another type of a sentence contains an attribute in addition. The attribute extends the subject or the object in the sentence. Examples of attributes are sizes, colors, i.e. "small", "big", "red", "blue", etc.

Table 1 Shortcuts and full URIs

Shortcuts	URI
ISA	http://en.wikipedia.org/wiki/Is-a
subject	http://en.wikipedia.org/wiki/Subject_(grammar)
predicate	http://en.wikipedia.org/wiki/Predicate_(grammar)
object	http://en.wikipedia.org/wiki/Object_(grammar)
attribute	http://en.wikipedia.org/wiki/Is-a
Prince Charles	http://en.wikipedia.org/wiki/Charles,_Prince_of_Wales
Elizabeth II	http://en.wikipedia.org/wiki/Elizabeth_II
foaf:knows	http://xmlns.com/foaf/0.1/knows
is	http://en.wikipedia.org/wiki/Is
young	http://en.wiktionary.org/wiki/young
Prince William	http://en.wikipedia.org/wiki/Prince_William
likes	http://en.wikipedia.org/wiki//Like
football	http://en.wikipedia.org/wiki/Football_(soccer)

Types of a negative declarative sentence are the same as for a positive declarative sentence. Negation in a sentence is related to a verb and it changes the verb to its opposite meaning. Examples of this type of a sentence are “I do not know John.”, “Michael does not like ice-cream.”, etc.

Identification of nodes and edges in an RDF graph is realized by the help of URIs, which can be quite long. RDF graph with full URIs would be confusing to the users. It is possible to use shortcuts for full URIs. In the following RDF graphs will be used shortcuts which are shown in Table 1.

4.2 Knowledge pattern for sentence in basic form

The first knowledge pattern presented here is knowledge pattern for conversion of a sentence in the basic form to RDF. This type of a sentence contains three parts – subject, predicate and object. In English, the order of parts is as follows – subject – verb – object. This sentence forms one RDF triple. The figure 5 shows this triple as a knowledge pattern. In the following, this pattern will be marked as KPS01.

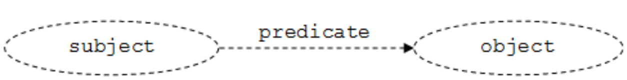

Figure 5. KPS01.

Let us consider that the root element of all following descriptions in RDF/XML will be the same. Its form will be as follows:

```
<rdf:RDF
   xmlns="http://www.r-miarka.net/kp.rdf#"
   xml:base="http://www.r-miarka.net/kp.rdf"
   xmlns:foaf="http://xmlns.com/foaf/0.1/"
   xmlns:wikien="http://en.wikipedia.org/
      wiki/"
   xmlns:rdfs="http://www.w3.org/2000/01/
      rdf-schema#"
   xmlns:rdf="http://www.w3.org/1999/02/
      22-rdf-syntax-ns#">
...
</rdf:RDF>
```

The representation of KPS01 in RDF/XML, which uses the vocabulary kp.rdf, is shown below.

```
<rdfs:Class rdf:about=
      "http://en.wikipedia.org/wiki/
      Subject_(grammar)">
   <rdfs:label xml:lang="en">Subject
      </rdfs:label>
   <rdfs:comment xml:lang="en">Subject in
      a sentence</rdfs:comment>
   <rdfs:subClassOf rdf:resource=
      "http://www.r-miarka.net/kp.rdf#
      KnowledgePatternClass"/>
   <kp:knowledgePatternCode>KPS01
      </kp:knowledgePatternCode>
   <wikien:Predicate_(grammar) rdf:resource=
      "http://en.wikipedia.org/wiki/
      Object_(grammar)"/>
</rdfs:Class>

<rdfs:Class rdf:about=
      "http://en.wikipedia.org/wiki/
      Object_(grammar)">
   <rdfs:label xml:lang="en">Object
      </rdfs:label>
   <rdfs:comment xml:lang="en">Object in
      a sentence</rdfs:comment>
```

```
   <rdfs:subClassOf rdf:resource=
      "http://www.r-miarka.net/kp.rdf
      #KnowledgePatternClass"/>
   <kp:knowledgePatternCode>KPS01
      </kp:knowledgePatternCode>
</rdfs:Class>

<rdf:Property rdf:about=
      "http://en.wikipedia.org/wiki/
      Predicate_(grammar)">
   <rdfs:label xml:lang="en">Predicate
      </rdfs:label>
   <rdfs:comment xml:lang="en">Predicate in
      a sentence</rdfs:comment>
   <rdfs:subPropertyOf rdf:resource=
      "http://www.r-miarka.net/kp.rdf#
      knowledgePatternProperty"/>
   <rdfs:domain rdf:resource=
      "http://en.wikipedia.org/wiki/
      Subject_(grammar)" />
   <rdfs:range rdf:resource=
      "http://en.wikipedia.org/wiki/
      Object_(grammar)" />
</rdf:Property>
```

The example of using knowledge pattern KPS01 is the sentence:

Prince Charles knows Elizabeth II.

Prince Charles is the subject of the sentence, Elizabeth II is the object and the predicate (relation) is knows. To represent of the predicate, property “knows” from vocabulary FOAF will be used. Morphism for terms from this sentence in a graph form is shown in next figure.

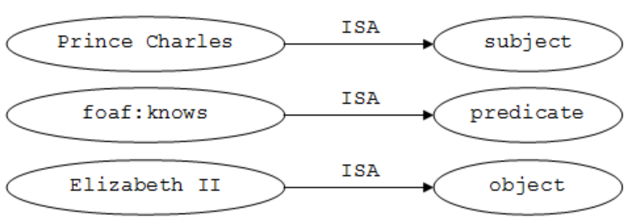

Figure 6. KPS01 - morphism.

After renaming the terms, the resulting RDF graph will look like that in next figure. This graph contains only one RDF triple.

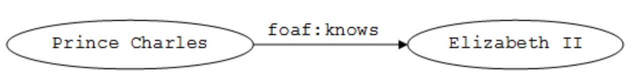

Figure 7. KPS01 - result.

Representation of pattern KPS02 in RDF/XML is shown below.

```
<rdf:Description rdf:about=
      "http://en.wikipedia.org/wiki/
      Prince_Charles">
   <kp:isa rdf:resource=
      "http://en.wikipedia.org/wiki/
      Subject_(grammar)" />
</rdf:Description>

<rdf:Description rdf:about=
      "http://xmlns.com/foaf/0.1/knows">
   <kp:isa rdf:resource=
      "http://en.wikipedia.org/wiki/
      Predicate_(grammar)" />
</rdf:Description>

<rdf:Description rdf:about=
      "http://en.wikipedia.org/wiki/
      Queen_Elizabeth_II">
   <kp:isa rdf:resource=
      "http://en.wikipedia.org/wiki/
      Object_(grammar)" />
</rdf:Description>
```

The result in RDF/XML contains one RDF triple:

```
<rdf:Description rdf:about=
      "http://en.wikipedia.org/wiki/
      Prince_Charles">
   <foaf:knows rdf:resource=
      "http://en.wikipedia.org/wiki/
      Queen_Elizabeth_II" />
</rdf:Description>
```

4.3 Knowledge pattern for sentence with attribute by subject

The second example of knowledge pattern is one for a sentence in the basic form which contains an attribute in addition. An attribute can extend the subject or the object in a sentence. An attribute can be represented by color (red, blue etc.) or size (small, big etc.). In this case, it will represent a sentence with an attribute by the subject. A graph representation of this knowledge pattern is shown in next figure.

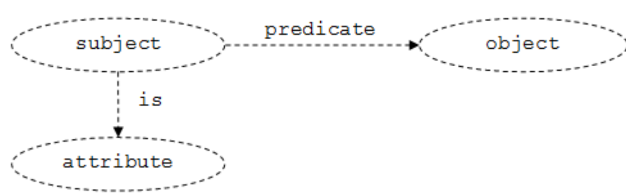

Figure 8. KPS02.

The RDF graph in the figure contains two RDF triples. The first triple represents a sentence in the basic form (see KPS01), the second triple represents an extension of the subject by an attribute. The predicate in this triple is general relation “is”. According to the type of attribute, it can be different. For color it could be “hasColor”, for size “hasSize”, etc. The node which represents the subject connects these two triples to the resulting RDF graph. For representation of nodes in the RDF graph are used URI identifiers. The node with a label “subject” has the same URI, so it can be displayed only once in the resulting RDF graph.

Representation of pattern KPS02 in RDF/XML is shown below.

```
<rdfs:Class rdf:about=
      "http://en.wikipedia.org/wiki/
      Subject_(grammar)">
   <rdfs:label xml:lang="en">Subject
      </rdfs:label>
   <rdfs:comment xml:lang="en">Subject in
      a sentence</rdfs:comment>
   <rdfs:subClassOf rdf:resource=
      "http://www.r-miarka.net/kp.rdf#
      KnowledgePatternClass"/>
   <kp:knowledgePatternCode>KPS02
      </kp:knowledgePatternCode>
   <wikien:Predicate_(grammar) rdf:resource=
      "http://en.wikipedia.org/
      wiki/Object_(grammar)"/>
   <wikien:Is rdf:resource=
      "http://en.wikipedia.org/wiki/
      Attribute"/>
</rdfs:Class>

<rdfs:Class rdf:about=
      "http://en.wikipedia.org/wiki/
      Object_(grammar)">
   <rdfs:label xml:lang="en">Object
      </rdfs:label>
   <rdfs:comment xml:lang="en">Object in
      a sentence</rdfs:comment>
   <rdfs:subClassOf rdf:resource=
      "http://www.r-miarka.net/kp.rdf#
      KnowledgePatternClass"/>
   <kp:knowledgePatternCode>KPS02
      </kp:knowledgePatternCode>
</rdfs:Class>

<rdf:Property rdf:about=
      "http://en.wikipedia.org/wiki/
      Predicate_(grammar)">
   <rdfs:label xml:lang="en">Predicate
      </rdfs:label>
   <rdfs:comment xml:lang="en">Predicate in
      a sentence</rdfs:comment>
   <rdfs:subPropertyOf rdf:resource=
      "http://www.r-miarka.net/kp.rdf#
      knowledgePatternProperty"/>
   <rdfs:domain rdf:resource=
      "http://en.wikipedia.org/wiki/
      Subject_(grammar)" />
   <rdfs:range rdf:resource=
      "http://en.wikipedia.org/wiki/
      Object_(grammar)" />
</rdf:Property>

<rdfs:Class rdf:about=
      "http://en.wikipedia.org/wiki/
      Attribute">
   <rdfs:label xml:lang="en">Attribute
      </rdfs:label>
   <rdfs:comment xml:lang="en">Attribute by
      a subject</rdfs:comment>
   <rdfs:subClassOf rdf:resource=
      "http://www.r-miarka.net/kp.rdf#
      KnowledgePatternClass"/>
   <kp:knowledgePatternCode>KPS02
      </kp:knowledgePatternCode>
</rdfs:Class>

<rdf:Property rdf:about=
      "http://en.wikipedia.org/wiki/Is">
   <rdfs:label xml:lang="en">Is</rdfs:label>
   <rdfs:comment xml:lang="en">
      Verb is</rdfs:comment>
   <rdfs:subPropertyOf rdf:resource=
      "http://www.r-miarka.net/kp.rdf#
      knowledgePatternProperty"/>
   <rdfs:domain rdf:resource=
      "http://en.wikipedia.org/wiki/
      Subject_(grammar)" />
   <rdfs:range rdf:resource=
      "http://en.wikipedia.org/wiki/
      Attribute" />
</rdf:Property>
```

An example of use of this knowledge pattern is sentence

Young Prince William likes football.

An attribute of this sentence is “young”. Morphism for this sentence in a graph form is shown in next figure.

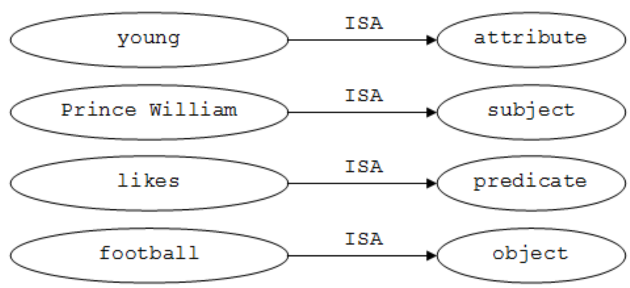

Figure 9. KPS02 - morphism.

Result after renaming the terms is shown in figure 10.

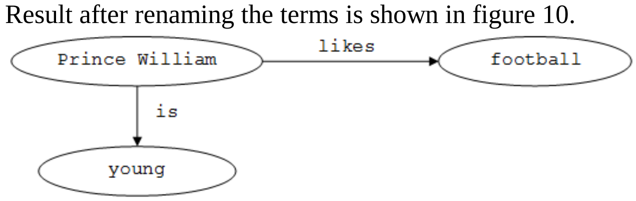

Figure 10. KPS02 - result.

The morphism for the given sentence in RDF/XML will look as follows:

```
<rdf:Description rdf:about=
      "http://en.wiktionary.org/wiki/young">
   <kp:isa rdf:resource=
      "http://en.wikipedia.org/wiki/
      Attribute" />
</rdf:Description>
```

```
<rdf:Description rdf:about=
      "http://en.wikipedia.org/wiki/
      Prince_William">
   <kp:isa rdf:resource=
      "http://en.wikipedia.org/wiki/
      Subject_(grammar)" />
</rdf:Description>

<rdf:Description rdf:about=
      "http://en.wikipedia.org/wiki/Like">
   <kp:isa rdf:resource=
      "http://en.wikipedia.org/wiki/
      Predicate_(grammar)" />
</rdf:Description>

<rdf:Description rdf:about=
      "http://en.wikipedia.org/wiki/
      Football_(soccer)">
   <kp:isa rdf:resource=
      "http://en.wikipedia.org/wiki/
      Object_(grammar)" />
</rdf:Description>
```

Result after using morphism is shown further. Both triples are connected together.

```
<rdf:Description rdf:about=
      "http://en.wikipedia.org/wiki/
      Prince_William">
   <wikien:Like rdf:resource=
      "http://en.wikipedia.org/wiki/
      Football_(soccer)"/>
   <wikien:Is rdf:resource=

   "http://en.wiktionary.org/wiki/young"/>
</rdf:Description>
```

5. Conclusions

Knowledge patterns can help with reusing knowledge. Knowledge patterns can be divided to two groups – Top-level knowledge patterns and Domain knowledge patterns. This article contains the proposal of representation of knowledge patterns in RDF(S) – in a graph form (RDF graph) and in a text form (RDF/XML). The usage of representation in RDF(S) enables to use knowledge patterns in the Semantic web (also called Web of data or Data web). Two examples of knowledge patterns and their use are a part of this article. The future work will be aimed at finding other Top-level knowledge patterns and at finding Domain knowledge patterns.

References

[1] P. Clark, J. Thompson and B. Porter, "Knowledge patterns," In *Handbook on Ontologies*, S. Staab and R. Studer, Eds. Berlin: Springer-Verlag, 2004, ISBN 3-540-40834-7, pp. 191–207.
[2] D. Crystal, "The Cambridge Encyclopedia of Language (2nd edition ed.)", Cambridge: Cambridge University Press, 1997, ISBN 0-521-55967-7.
[3] M. S. Dryer, "Order of Subject, Object, and Verb", http://linguistics.buffalo.edu/people/faculty/dryer/dryer/DryerWalsSOVNoMap.pdf
[4] A. Lukasová, M. Vajgl and M. Žáček, "Reasoning in RDF graphic formal system with quantifier," *Proceedings of the International Multiconference on Computer Science and Information Technology*, 2010, ISBN 978-83-60810-22-4, pp. 67–72.
[5] S. Staab, M. Erdmann, A. Maedche and S. Decker "Ontologies in RDF(S)," [online] http://www.ida.liu.se/ext/etai/received/semaweb/010/paper.pdf.
[6] W3C: Resource Description Framework (RDF): Concepts and Abstract Syntax, http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/
[7] W3C: RDF Primer, http://www.w3.org/TR/2004/REC-rdf-primer-20040210/

Rostislav Miarka graduated fromt the University of Ostrava (Czech Republic). The focus of his dissertation is a formal representation of knowledge patterns in ontological languages. The topic of the PhD thesis corresponds to the content of the article. He is an author more than 10 scientific publications.