Development of a knowledge base based on context analysis of external information resources

N Yarushkina[1], A Filippov[1] and V Moshkin[1]

[1]Ulyanovsk State Technical University, Severny Venetz str. 32, Ulyanovsk, Russia, 432027

Abstract. The article describes the process of developing a knowledge base (KB). The content of KB is formed as a result of the analysis of the contexts of external information resources. In this case, the context is a certain "point of view" on the problem area (PrA) and its features. A graph database (DB) Neo4j is used as the basis for storing the contents of the KB in the form of an ontology. An attempt is made to implement the mechanism of inference by the contents of a graph database. The mechanism is used to dynamically generate the screen forms of the user interface to simplify the work with the KB. This article also describes the method of extension of KB based on the content of the wiki-resources and relational databases.

1. Introduction

Post-industrial society operates with huge volumes of information both in everyday and professional activities. A large amount of information causes difficulties in making decisions within the framework of rigid time constraints.

A variety of software automation of human activities are used to solve this problem. However, it is necessary to adapt them to the specifics of a particular problem area (PrA) and its contexts for the effective operation of these tools [1, 2, 7, 10, 18, 19, 20].

Thus, "trained" automation software solves the tasks more efficiently, but they require considerable resources (human and temporary) for training.

In this paper, an attempt is made to construct a KB. The content of the KB is an applied ontology. The basic requirements for KB are [26]:

- adaptation to the specifics of PrA based on contexts;
- reliability and speed of ontology storage;
- the presence of a mechanism of logical inference;
- availability of tools to simplify work with the KB for unprepared users;
- availability of mechanisms for importing data from external information resources.

As you can see from Figure 1, the KB consists of the following subsystems:

(i) Ontology store:
- Neo4j [12];
- content management module;
- ontology import/export module.

(ii) Inference subsystem:

- inference module.

(iii) A subsystem for interaction with users:
- screen forms generation module.

(iv) A subsystem for importing data from wiki-resources:
- a module for importing data from wiki-resources.

(v) A subsystem for importing data from relational databases:
- a module for importing data from relational databases.

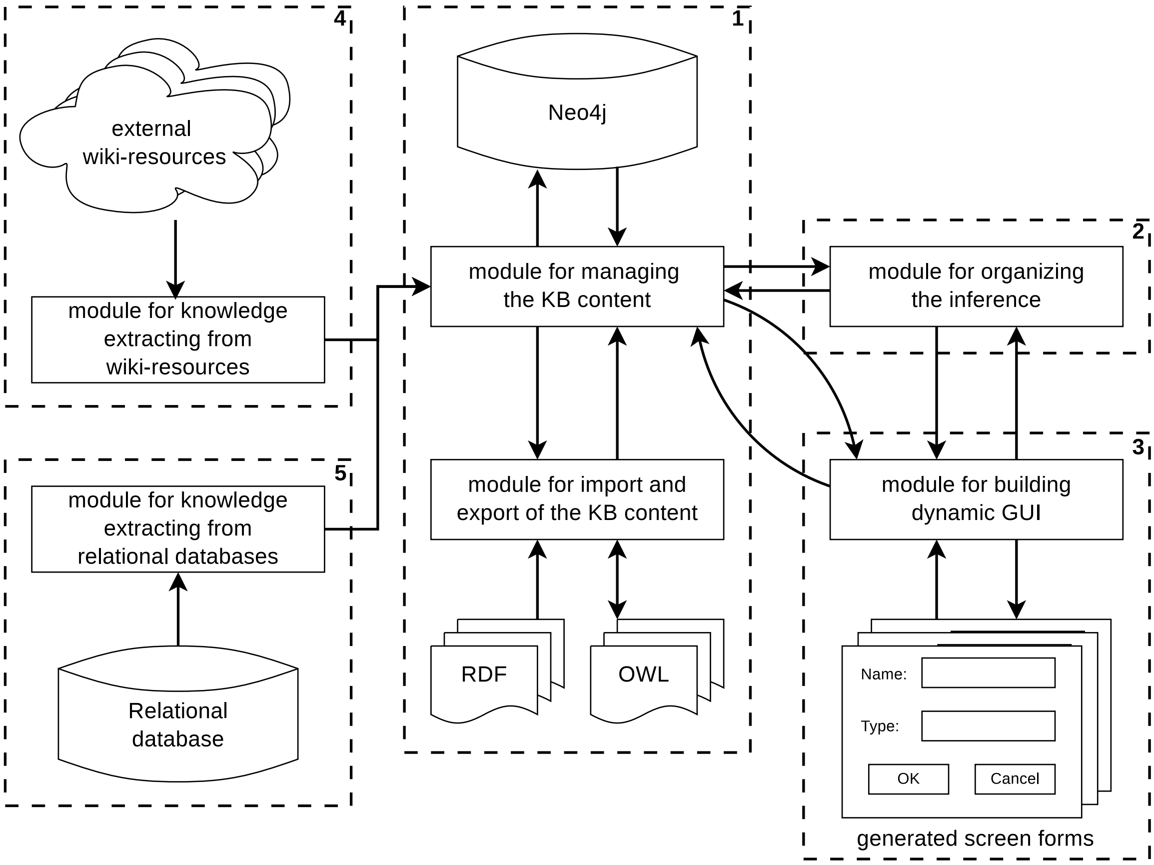

Figure 1. Knowledge base architecture.

2. The organization of the ontology store of KB

Ontology is a model of the representation of the PrA in the form of a semantic graph [9].

Graph-oriented database management system (Graph DBMS) Neo4j is the basis of the ontology store for KB. Neo4j is currently one of the most popular graph databases and has the following advantages:

(i) Having a free community version.

(ii) Native format for data storage.

(iii) One copy of Neo4j can work with graphs containing billions of nodes and relationships.

(iv) The presence of a graph-oriented query language Cypher.

(v) Availability of transaction support.

Neo4j was chosen to store the description of the PrA in the form of an applied ontology, since the ontology is actually a graph. In this case, it is only necessary to limit the set of nodes and graph relations into which ontologies on RDF and OWL will be translated.

The context of an KB is some state of content of KB, obtained during versioning or building a content of KB using different "points of view" [6, 8].

Figure 2 shows an example of the translation of the owl representation of ontology of family relations into the entities of the KB.

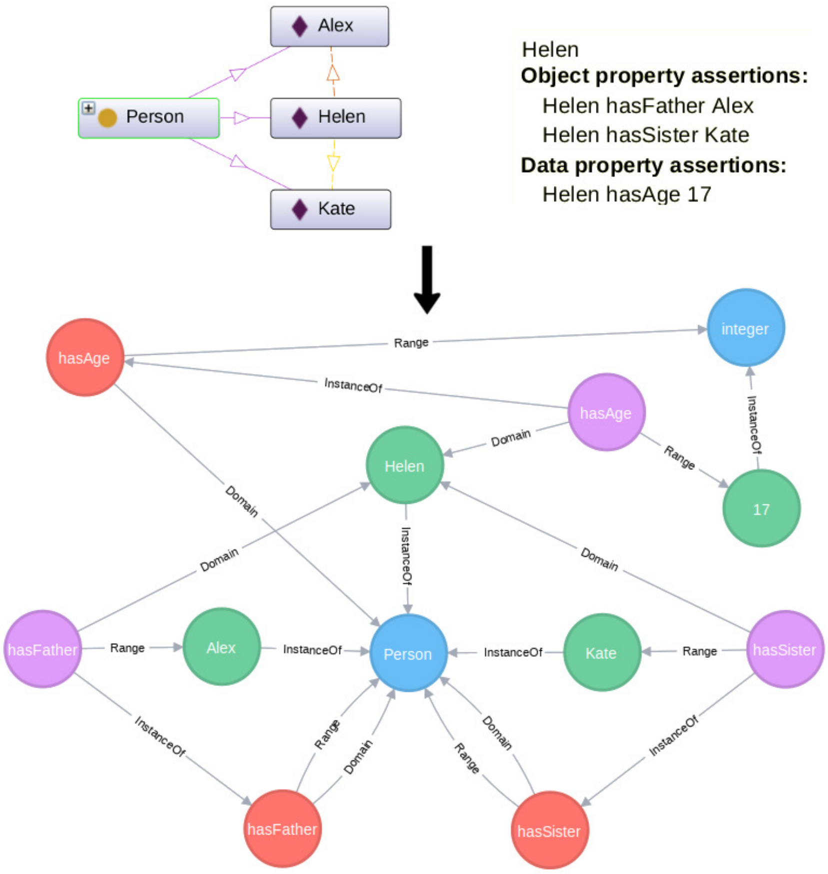

Figure 2. Example of the translation of the owl representation of ontology of family relations into the content of the KB.

Formally, the content of the KB can be represented by the following equation:

$$O = \langle T, C^{T_i}, I^{T_i}, P^{T_i}, S^{T_i}, F^{T_i}, R^{T_i} \rangle, i = \overline{1, t}, \tag{1}$$

where t is a number of the KB contexts,
$T = \{T_1, T_2, \ldots, T_t\}$ is a set of KB contexts,
$C^{T_i} = \{C_1^{T_i}, C_2^{T_i}, \ldots, C_n^{T_i}\}$ is a set of KB classes within the i-th context,
$I^{T_i} = \{I_1^{T_i}, I_2^{T_i}, \ldots, I_n^{T_i}\}$ is a set of KB objects within the i-th context,
$P^{T_i} = \{P_1^{T_i}, P_2^{T_i}, \ldots, P_n^{T_i}\}$ is a set of KB classes properties within the i-th context,
$S^{T_i} = \{S_1^{T_i}, S_2^{T_i}, \ldots, S_n^{T_i}\}$ is a set of KB objects states within the i-th context,
$F^{T_i} = \{F_1^{T_i}, F_2^{T_i}, \ldots, F_n^{T_i}\}$ is a set of the logical rules fixed in the KB within the i-th context,

R^{T_i} is a set of KB relations within the i-th context defined as:

$$R^{T_i} = \{R_C^{T_i}, R_I^{T_i}, R_P^{T_i}, R_S^{T_i}, R_F^{T_i}\},$$

where $R_C^{T_i}$ is a set of relations defining hierarchy of KB classes within the i-th context,
$R_I^{T_i}$ is a set of relations defining the "class-object" KB tie within the i-th context,
$R_P^{T_i}$ is a set of relations defining the "class-class property" KB tie within the i-th context,
$R_S^{T_i}$ is a set of relations defining the "object-object state" KB tie within the i-th context,
$R_F^{T_i}$ is a set of relations generated on the basis of logical KB rules in the context of i-th context.

Principles similar to the paradigm of object-oriented programming are at the basis of the content of the KB:

- KB classes are concepts of the PrA;
- classes can have properties, the child-class inherits properties of the parent class;
- objects of KB describe instances of the concepts of the PrO;
- specific values for the properties of objects inherited from the parent class are determined by the states;
- logical rules are used to implement the functions of inference by the content of KB.

3. The inference on the contents of KB

The inference is the process of reasoning from the premises to the conclusion. Reasoners are used to implement the function of inference. Reasoners form logical consequences on the basis of many statements, facts and axioms. The most popular at the moment reasoners are [5, 17]:

- Pellet;
- FaCT++;
- Hermit;
- Racer, etc.

These reasoners are actively used in the development of intelligent software. However, Neo4j does not assume the possibility of using similar default reasoners. Thus, there is a need to develop a mechanism for inference based on the content of a KB [3, 4].

Currently the Semantic Web Rule Language (SWRL) is used to record logical rules [24].

These SWRL rules describe the conditions under which object a has "nephew-uncle" relation with object c. Formally the logical rule of the KB is:

$$F^{T_i} = \langle A^{Tree}, A^{SWRL}, A^{Cypher}\rangle,$$

where T_i is the i-th context of the the KB, A^{Tree} is the tree-like representation of a logical rule F^{T_i}, A^{SWRL} is the SWRL representation of the logical rule F^{T_i}, A^{Cypher} is the Cypher representation of the logical rule F^{T_i}.

The tree-view A^{Tree} of a logical rule F^{T_i} is:

$$A^{Tree} = \langle Ant, Cons\rangle,$$

where $Ant = Ant_1 \Theta Ant_2 \Theta \ldots \Theta Ant_n$ is the antecedent (condition) of the logical rule F^{T_i};
$\Theta \in \{AND, OR\}$ is a set of permissible logical operations between antecedent atoms;
$Cons$ is the consequent (consequence) of a logical rule F^{T_i}.

Figure 3 shows an example of a tree-like representation of two logical rules for the ontology of family relations. That rules describes the father-child relationships.

The tree-like logical rule is translated into the following SWRL:

```
hasFather(?a,?b) => hasChild(?b,?a)
hasSister(?c,?a) & hasFather(?c,?b) => hasChild(?b,?a)
```

and the following Cypher view:

```
MATCH (s1:Statement{name: "hasChild", lr: true})
MATCH (r1a)<-[:Domain]-(:Statement{name:"hasFather"})-[:Range]->(r1b)
MERGE (r1b)-[:Domain]->(s1)
MERGE (r1a)-[:Range]->(s1)

MATCH (s1:Statement{name: "hasChild", lr: true})
MATCH (r2c)<-[:Domain]-(:Statement{name:"hasSister"})-[:Range]->(r2a)
MATCH (r2c)<-[:Domain]-(:Statement{name:"hasFather"})-[:Range]->(r2b)
MERGE (r2b)-[:Domain]->(s1)
MERGE (r2a)-[:Range]->(s1)
```

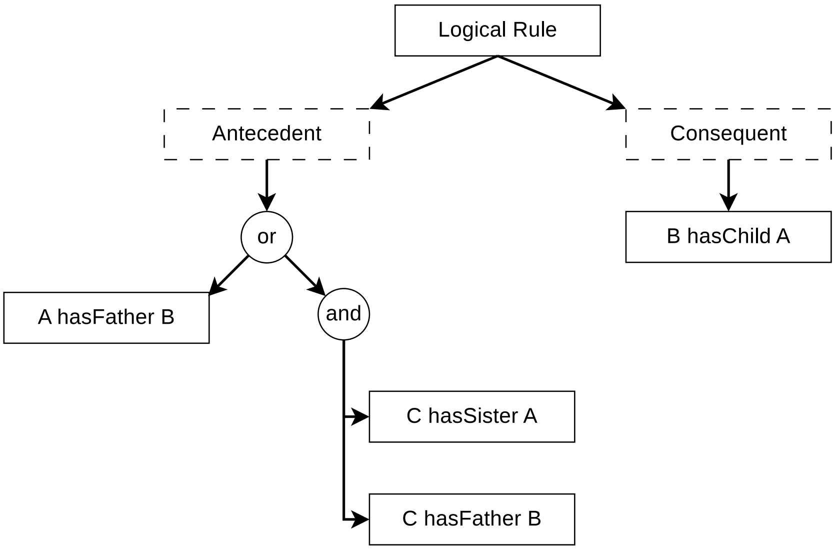

Figure 3. Example of a tree-like representation of a logical rule.

Thus, the rules are translated into their tree-view when imported into the KB of logical rules in the SWRL language.

The presence of a tree-like representation of a logical rule allows to form both a SWRL-representation of a logical rule and a Cypher-representation based on it.

Relations of a special type are formed by using Cypher to represent the logical rule between entities of the KB. Figure 4 shows the content of KB after executing the Cypher queries that were built for the logical rule shown in Figure 3. These relations correspond to the antecedent atoms of the logical rule. Formed relationships provide the inference from the contents of the KB.

4. Building a Graphical User Interface based on the contents of a KB

The dynamic graphical user interface (GUI) mechanism is used to simplify the work with KB of untrained users and control of user input [11, 13, 21].

You need to map the KB entities to the GUI elements to build a GUI based on the contents of the KB. Formally, the GUI model can be represented as follows:

$$UI = \langle L, C, I, P, S \rangle, \tag{2}$$

where $L = \{L_1, L_2, \ldots, L_n\}$ is a set of graphical GUI components (for example, ListBox, TextBox, ComboBox, etc.), $C = \{C_1, C_2, \ldots, C_n\}$ is a set of KB classes, $I = \{I_1, I_2, \ldots, I_n\}$ is a set of KB objects, $P = \{P_1, P_2, \ldots, P_n\}$ is a set of properties of KB classes, $S = \{S_1, S_2, \ldots, S_n\}$ is a set of states of KB objects.

Figure 4. The result of executing Cypher queries for logical rule.

The following function is used to build a GUI based on content of KB:

$$\phi(O) : \{C^O, I^O, P^O, S^O, F^O, R^O\}^{T_i} \rightarrow \{L^{UI}, C^{UI}, I^{UI}, P^{UI}, S^{UI}\},$$

where $\{C^O, I^O, P^O, S^O, F^O, R^O\}^{T_i}$ is a set of entities of KB represented by expression 1 within the i-th context;
$\{L^{UI}, C^{UI}, I^{UI}, P^{UI}, S^{UI}\}$ is a set of GUI entities of KB represented by the expression 2.

Thus, the contents of the KB are mapped to set of GUI components. This makes it easier to work with KB for a user who does not have skills in ontological analysis and knowledge engineering. It also allows you to monitor the logical integrity of the user input, which leads to a reduction in the number of potential input errors.

5. Extracting knowledge from wiki-resources

At present, wiki-technologies are used to organize corporate KB. It is necessary to solve the task of knowledge extracting from wiki-resources [14, 15, 16, 23, 27]. Table 1 contains the result of mapping the KB entities to the wiki-resource entities [22]. Thus, it becomes possible to import the structure of external wiki resources for initial filling of the KB contents.

Table 1. The correspondence between the wiki-resource entities and the entities of KB.

The entities of knowledge base	The entities of wiki-resource
Class	Category
Subclass	Subcategory
Object	Page
Class properties	The infobox elements (properties)
Object states	The infobox elements (values)
Relations	Hyperlinks

Also, a content of KB can be built on the basis of an analysis of the content of wiki-resources pages. In this work the Syntaxnet [22] framework to construct a syntactic tree $Synt$ of content of wiki-resources pages is used. Further, using a set of rules $Rule^{Synt}$, a syntax tree $Synt$ is translated into entities of KB.

Formally the functions of translating a syntactic tree into entities of KB:

$$\phi^{Struct}(Synt) : \{N_{Synt}, Rule^{Struct}_{Synt}\} \rightarrow \{C^O, P^O, R^O_P\}^{T_i},$$
$$\phi^{Content}(Synt) : \{N_{Synt}, Rule^{Content}_{Synt}\} \rightarrow \{I^O, S^O, R^O_I, R^O_S\}^{T_i},$$

where N_{Synt} is a set of nodes of the syntactic tree $Synt$,
$Rule^{Struct}_{Synt}$ is a set of rules to translating nodes of syntactic into structure entities of the KB,
$Rule^{Content}_{Synt}$ is a set of rules to translating nodes of syntactic into content entities of the KB,
$\{C^O, P^O, R^O_P\}^{T_i}$ is a set of structure entities of the KB within the context T_i (eq. 1),
$\{I^O, S^O, R^O_I, R^O_S\}^{T_i}$ is a set of content entities of the KB within the context T_i (eq. 1).

Formally the rules to translating nodes of syntactic into entities of the KB:

$$Rule^{Struct}_{Synt} = \left(N^{Synt}_1, N^{Synt}_2, \ldots, N^{Synt}_i, \ldots, N^{Synt}_n\right) \rightarrow \{C^O, P^O, R^O_P\},$$
$$Rule^{Content}_{Synt} = \left(N^{Synt}_1, N^{Synt}_2, \ldots, N^{Synt}_i, \ldots, N^{Synt}_m\right) \rightarrow \{I^O, S^O, R^O_I, R^O_S\},$$

where N^{Synt}_i is the i-th node of syntactic tree.

Thus, it becomes possible to extract knowledge from the structure of wiki-resource and contents of wiki-resource pages and present the extracted knowledge as a content of KB.

6. Extracting knowledge from relational databases

Relational databases are widely used for data storing and contains subject area description in the form of interconnected tables. Nowadays, researchers of various scientific groups are involved in solving the problem of extracting knowledge from relational databases.

The relational data model can be represented as the following expression:

$$RDM = (E, R),$$

where $E = \{E_1, E_2, \ldots, E_n$ is a set of database tables (entities),
$R = \{R_1, R_2, \ldots, R_i, \ldots R_n$ is a set of relationships between database tables:

$$R_i = E_j \frac{F(x)}{G(x)} E_k,$$

where E_j, E_k are database entities;
$F(x)$ is the relationship between entity E_j and entity E_k,
$G(x)$ is the relationship between entity E_k and entity E_j.

Scope of functions $F(x)$ and $G(x)$ are U – single relationship and N – multiply relationship.

For mapping of relational database structure with KB structure special functions are used:

$$\phi^{Struct}(RDM) : \{E^{RDM}, R^{RDM}\} \to \{C^O, P^O, R_P^O\}^{T_i},$$
$$\phi^{Content}(RDM) : \{E^{RDM}, R^{RDM}\} \to \{I^O, S^O, R_I^O, R_S^O\}^{T_i},$$

where $\{E^{RDM}, R^{RDM}\}$ is a set of entities of relational database and relationships between them,
$\{C^O, P^O, R_P^O\}^{T_i}$ is a set of structure entities of the KB within the context T_i (eq. 1),
$\{I^O, S^O, R_I^O, R_S^O\}^{T_i}$ is a set of content entities of the KB within the context T_i (eq. 1).

Importing data from a relational database to the KB were finish after mapping the structure of the relational database to the set of structure entities $\{C^O, P^O, R_P^O\}^{T_i}$ of the KB ends. Set of content entities of KB $\{I^O, S^O, R_I^O, R_S^O\}^{T_i}$ are created during the import of data basis from the relational database (*row set*) to the T_i context. Table 2 contains a comparison of KB entities with relational database entities.

Table 2. The correspondence between the relational database entities and the entities of KB

The entity of knowledge base	The entity of relational database
Class	Table
Object	Table row
Class properties and Relations	Relations between tables, table columns
Object states	Content of cells

Thus, it becomes possible to extract knowledge from the contents of relational databases and present the extracted knowledge as a content of KB.

7. Conclusion

Thus, the use of KB stored in the Graph DBMS in the decision support process presupposes the existence of a certain set of mechanisms:

- organization of inference on the content of KB by translating SWRL-rules into Cypher-structures;
- building a graphical user interface based on the contents of KB;
- automated import of knowledge from structure and content of wiki-resources;
- automated import of knowledge from relational databases.

These mechanisms allow to automate the learning process of KB and simplify the work of specialists with KB. The application of a contextual approach to the storage of knowledge raises the effectiveness of the use of subject ontologies, allowing to adapt the KB to the characteristics of the PrA and to the requirements of specialists. This approach provides them with a tool that is convenient in a software dynamically changeable depending on the contents of the KB.

8. References

[1] Berant J, Chou A, Frostig R and Liang P 2013 Semantic parsing on freebase from question-answer pairs *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)* 1533-1544

[2] Bianchini D, De Antonellis V, Pernici B and Plebani P 2005 Ontology-based methodology for e-service discovery *Information Systems* **31** 361380
[3] Bobillo F and Straccia U 2008 FuzzyDL: an expressive fuzzy description logic reasoner *Proceedings of the 17th IEEE International Conference on Fuzzy Systems* 923-930
[4] Bobillo F and Straccia U 2010 Representing fuzzy ontologies in OWL 2 *Proceedings of the 19th IEEE International Conference on Fuzzy Systems* 2695-2700
[5] Dentler K, Cornet R, Aten Teije and N de Keizer 2011 Comparison of reasoners for large ontologies in the OWL 2 EL profile *Semant. web* **2** 7187
[6] Falbo R A, Quirino G K, Nardi J C, Barcellos M P, Guizzardi G and Guarino N 2016 An ontology pattern language for service modeling *Proceedings of the 31st Annual ACM Symposium on Applied Computing* 321-326
[7] Farid D M, Al-Mamun M A, Manderick B and Nowe A 2016 An adaptive rule-based classifier for mining big biological data *Expert Systems with Applications* **64** 305316
[8] Gao M and Liu C 2005 Extending OWL by fuzzy description logic *Proceedings of the 17th IEEE International Conference on Tools with Artificial Intelligence* 562-567
[9] Guarino N and Musen M A 2015 Ten years of Applied Ontology *Applied Ontology* **10** 169170
[10] Guizzardi G, Guarino N, Almeida J P A 2016 Ontological Considerations About the Representation of Events and Endurants in Business Models *International Conference on Business Process Management* 20-36
[11] Hattori S and Takama Y 2014 Recommender System Employing Personal-Vallue-Based User Model *J Adv. Comput. Intell. Intell. Inform.* **18** 157165
[12] *Neo4j* (Access mode: https://neo4j.com/product) (14.05.2018)
[13] Ltifi H, Kolski C, Ayed M B and Alimi A M 2013 A human-centred design approach for developing dynamic decision support system based on knowledge discovery in databases *Journal of Decision Systems* **22** 6996
[14] Mikhaylov D V, Kozlov A P and Emelyanov G M 2015 An approach based on TF-IDF metrics to extract the knowledge and relevant linguistic means on subject-oriented text sets *Computer Optics* **39(3)** 429-438 DOI: 10.18287/0134-2452-2015-39-3-429-438
[15] Mikhaylov D V, Kozlov A P and Emelyanov G M 2016 Extraction of knowledge and relevant linguistic means with efficiency estimation for the formation of subject-oriented text sets *Computer Optics* **40(4)** 572-582 DOI: 10.18287/2412-6179-2016-40-4-572-582
[16] Mikhaylov D V, Kozlov A P and Emelyanov G M 2017 An approach based on analysis of n-grams on links of words to extract the knowledge and relevant linguistic means on subject-oriented text sets *Computer Optics* **41(3)** 461-471 DOI: 10.18287/2412-6179-2017-41-3-461-471
[17] *Pellet Framework* (Access mode: http://github.com/stardog-union/pellet) (14.05.2018)
[18] Rajpathak D, Chougule R and Bandyopadhyay P 2012 A domain-specific decision support system for knowledge discovery using association and text mining *Knowledge and Information Systems* **31** 405432
[19] Renu R S, Mocko G and Koneru A 2013 Use of Big Data and Knowledge Discovery to Create Data Backbones for Decision Support Systems *Procedia Computer Science* **20** 446453
[20] Rubiolo M, Caliusco M L, Stegmayer G, Coronel M, Fabrizi M G 2012 Knowledge discovery through ontology matching: An approach based on an Artificial Neural Network model *Information Sciences* **194** 107-119
[21] Ruy F B, Reginato C C, Santos V A, Falbo R A and Guizzardi G 2015 Ontology Engineering by Combining Ontology Patterns *34th International Conference on Conceptual Modeling* 173-186
[22] Shestakov V K 2011 Development and maintenance of information systems based on ontology and Wiki-technology *Advanced Methods and Technologies, Digital Collections* 299-306 (in Russian)

[23] Suchanek F M, Kasneci G and Weikum G 2007 YAGO: A Core of Semantic Knowledge Unifying WordNet and Wikipedia *Proceedings of the 16th International Conference on World Wide Web* 697706
[24] *SWRL: A Semantic Web Rule Language Combining OWL and RuleML* (Access mode: https://www.w3.org/Submission/SWRL) (14.05.2018)
[25] *SyntaxNet: Neural Models of Syntax* (Access mode: https://github.com/tensorflow/models/tree/master/research/syntaxnet) (14.05.2018)
[26] Yarushkina N, Filippov A and Moshkin V 2017 Development of the Unified Technological Platform for Constructing the Domain Knowledge Base Through the Context Analysis *Creativity in Intelligent Technologies and Data Science* 6272
[27] Zarubin A, Koval A, Filippov A and Moshkin V 2017 Application of Syntagmatic Patterns to Evaluate Answersto Open-Ended Questions *Creativity in Intelligent Technologies and Data Science* 150162

Acknowledgments
This work was financially supported by the Russian Foundation for Basic Research (Grant No. 16-47-732054).