Development of a Technological Platform for Knowledge Discovery

Nadezhda Yarushkina, Aleksey Filippov(✉), and Vadim Moshkin

Ulyanovsk State Technical University, Severny Venets Str., 32,
432027 Ulyanovsk, Russian Federation
{jng,al.filippov}@ulstu.ru, postforvadim@ya.ru
http://www.ulstu.ru/

Abstract. The paper considers the architecture of the technological platform designed for construction of the knowledge base (KB) by integrating a set of logical rules with fuzzy ontologies. The KB represents the storage of knowledge and contexts of different problem areas (PrA). The PrA ontology context is a specific state of the KB content that an expert can choose from a set of available states. The state is a result of either versioning or constructing the KB content from different points of views. Also, the paper describes the application of the KB in the inference of expert recommendations on solving the problem situations that occurred in the process of local area network functioning.

Keywords: Knowledge base · Ontology · Context · Problem area · Fuzzy ontology · Inference

1 Introduction

Modern organizations have a large amount of accumulated knowledge presented in the form of various corporate knowledge bases. Tools for creating wiki-resources are usually used for the organization of corporate knowledge bases. Technologies for creating wiki-resources are user-oriented and make it easy to create corporate knowledge bases of any complexity. Technologies for creating wiki-resources do not have a convenient way of querying knowledge for third-party software systems. Technologies for creating wiki-resources cannot discover new knowledge based on existing ones [1–4]. Thus, the development of a technological platform (TP) is necessary to solve the following tasks:

- the TP should not require additional skills and knowledge from the user;
- the TP should provide the programmer the familiar and easy to use data access mechanism;
- the TP should provide an inference mechanism to implement knowledge discovery (KD) functionality;

The study was supported by the Ministry of Science and Higher Education of the Russian Federation in framework of projects 2.4760.2017/8.9 and 2.1182.2017/4.6.

© Springer Nature Switzerland AG 2019
S. Misra et al. (Eds.): ICCSA 2019, LNCS 11619, pp. 533–544, 2019.
https://doi.org/10.1007/978-3-030-24289-3_39

– the TP should provide functions to the automation of the process of obtaining essential knowledge about problem area (PrA) in the internal knowledge base (KB).

At the moment, a lot of researchers use the ontological approach for the organization of the knowledge bases of expert and intelligent systems: Bobillo, Straccia [5,6], Gao, Liu [7], Bianchini [8], Guarino [9], Guizzardi [10], Falbo [11], Stumme [12], Gruber [13], Medche [14]. Ontology is a model for representing knowledge of the PrA. Graph-oriented database management system (Graph DBMS) Neo4j [15] was chosen to store the description of the PrA in the form of an applied ontology since the ontology is a graph. In this case, it is only necessary to limit the set of nodes and graph relations into which ontologies on OWL will be translated.

The inference is the process of reasoning from the premises to the conclusion. Reasoners [16,17] are used to implementing the function of inference and form logical consequences from many statements, facts, and axioms. Currently, the Semantic Web Rule Language (SWRL) is used to record logical rules for reasoners [18]. However, Neo4j does not assume the possibility of using default reasoners. Thus, there is a need to develop a mechanism for inference based on the content of a KB [5,6].

For successful translation, the wiki-resource into KB entities is required to assign the wiki-resource elements to TBox (structure) and ABox (content) of ontology [19].

2 The Model of the Fuzzy Domain KB Content with Contexts Support

Contexts of the KB represent the parts of ontology in space and time. Each space context is associated with a value from 0 to 1 defining the expert level of expertise in the part of the PrA. Time contexts allow using versioning of the PrA ontology and give an opportunity to monitor the dynamics of the ontology development. The fuzzy nature of the KB appears in the process of integration of contexts of the domain ontology.

The problem of developing the model of fuzzy domain KB with support of logical rules for inference is coming up. One of the KB main objectives is providing the mechanism for adapting the TP [20,21] to the concrete PrA with the use of methods of ontological analysis and data engineering.

Let the following definition represents the model of the KB content:

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

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, logical rules are used to implement the functions of inference by the content of KB,
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 the i-th context.

Figure 1 shows an example of the translation of the OWL representation of the ontology of family relations into the entities of the KB.

As seen in Fig. 1:

- OWL class "Person" was translated into the KB class with the same name;
- OWL individuals "Alex", "Helen", "Kate" and "17" was translated into the KB objects with same names;
- KB objects "Alex", "Helen" and "Kate" are objects of KB class "Person";
- KB object "17" is the object of built-in KB class "Integer";
- OWL data property "hasAge" was translated into the KB property with the same name;
- OWL object properties "hasFather" and "hasSister" was translated into the KB properties with the same names;
- OWL data property assertion "Helen hasAge 17" was translated into the KB state with the same name, the range of this state is "Helen", the domain of this state is "17";
- OWL object property assertion "Helen hasFather Alex" was translated into the KB state with the same name, the range of this state is "Helen", the domain of this state is "Alex";
- OWL object property assertion "Helen hasSister Kate" was translated into the KB state with the same name, the range of this state is "Helen", the domain of this state is "Kate".

2.1 The Inference on the Contents of KB

Formally the logical rule of the KB is:

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

where T_i is a i-th context of the KB,

A^{Tree} is a tree-like representation of the logical rule F^{T_i},

A^{SWRL} is a SWRL representation of the logical rule F^{T_i},

A^{Cypher} is a Cypher representation of the logical rule F^{T_i}.

A tree-like representation A^{Tree} of the 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 a 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 a consequent (consequence) of the logical rule F^{T_i}.

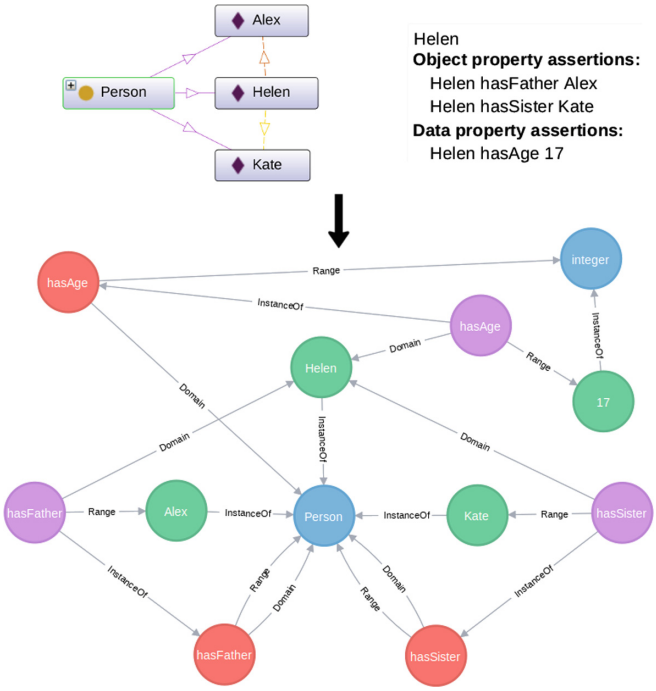


Fig. 1. Example of the translation of the OWL representation of ontology of family relations into the content of the KB.

The rules in the SWRL language are translated into their tree-like representations when imported into the KB of logical rules. The presence of a tree-like representation of a logical rule allows forming both an SWRL-representation of a logical rule and a Cypher-representation based on it.

For example, consider the process of translating the following SWRL rules into a tree-like representation (Fig. 2):

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

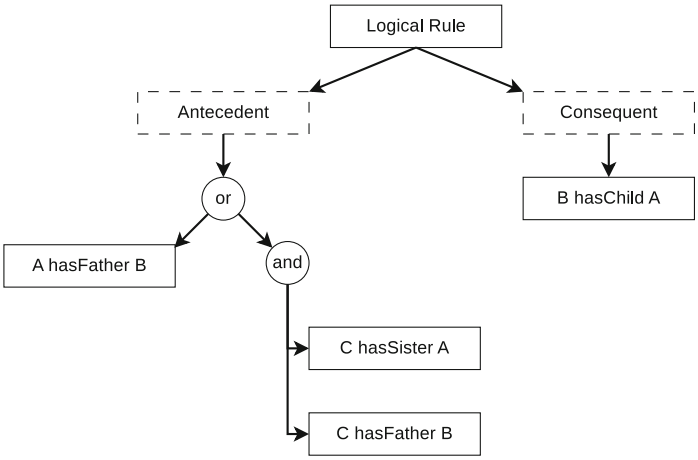


Fig. 2. Example of the tree-like representation of the logical rule.

The tree-like representation of the logical rule that presented in Fig. 2 is translated into the following Cypher representation:

```
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)
```

Relations of a particular type are formed by using the constructed Cypher queries to represent the logical rule between entities of the KB. Figure 3 shows the content of KB after executing the Cypher queries that were built for the tree-like representation of logical rule shown in Fig. 2. These relations correspond to the antecedent atoms of the logical rule. Formed relationships provide the inference from the contents of the KB.

As shown in Fig. 3, after the execution of the Cypher query from the Cypher representation of the logical rule (Fig. 2) the KB property with name “hasChild”



Fig. 3. The result of executing Cypher queries for the logical rule.

and the KB state with name "hasChild" were created. The range of created KB state "hasChild" is "Alex", the domains are "Helen" and "Kate".

3 Extracting Knowledge from Wiki-Resources

Let us represent the wiki-resource content by the following definition:

$$WIKI = \langle C, P, R \rangle, \tag{2}$$

where $C = \{C_1, C_2, \ldots, C_n\}$ is a set of wiki-resource categories,
$P = \{P_1, P_2, \ldots, P_i, \ldots, P_n\}$ is a set of wiki-resource pages defined as:

$$P_i = \langle N, I \rangle,$$

where N is a name of the i-th wiki-resource page P_i,
I is a infobox of the i-th wiki-resource page P_i defined as:

$$I = \{(K_1, V_1), (K_2, V_2), \ldots, (K_i, V_i), \ldots, (K_n, V_n)\},$$

where K_i is the i-th property of the infobox I of the wiki-resource page P_i,

V_i is a value of the i-th property K_i of the infobox I of the wiki-resource page P_i,

R is a set of relations between wiki-resource entities:

$$R = \{R_C, R_P, R_{CP}\},$$

where R_C is a set of relations defining a hierarchy of wiki-resource categories,
R_P is a set of relations defining the "page-page" tie (hyperlinks),
R_{CP} is a set of relations defining the "category-page" tie.

The following function is used to translate the wiki-resource content into KB content:

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

where $\{C^{WIKI}, P^{WIKI}, R^{WIKI}\}$ is a set of wiki-resource entities (Eq. 2),
$\{C^O, I^O, P^O, S^O, R^O\}^{T_i}$ is a set of KB entities within the context T_i (Eq. 1).

Table 1 contains the result of mapping the wiki-resource entities to the KB entities [22].

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

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

4 The Architecture of the Technological Platform

Figure 4 shows that the TP consists of the following subsystems:

1. Ontology storage:
   - Neo4j;
   - KB content management module;
   - module for import of fuzzy ontologies with a set of SWRL rules.
2. Inference subsystem:
   - inference module.
3. Subsystem for interaction with users:
   - screen forms generation module.
4. Subsystem for importing data from wiki-resources:
   - module for importing data from wiki-resources.

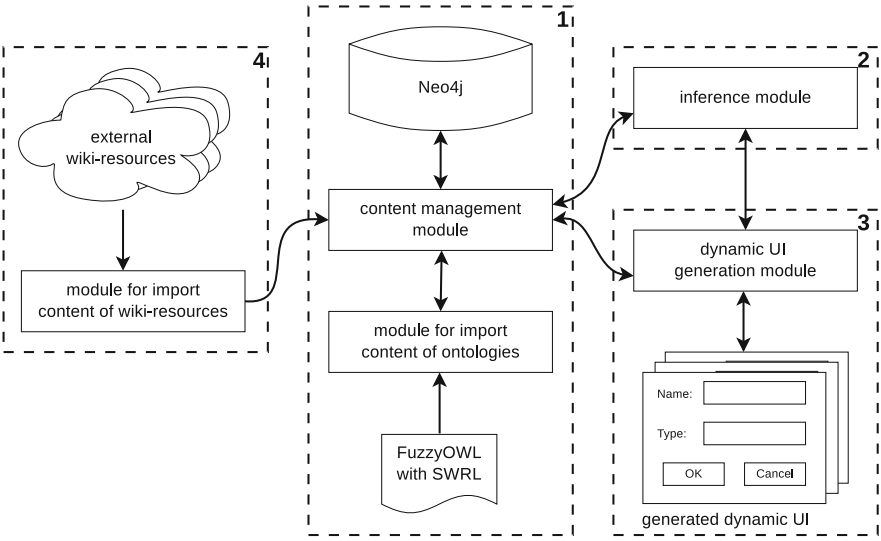


Fig. 4. Knowledge base architecture.

The dynamic graphical user interface (GUI) mechanism is used to simplify the work with KB of untrained users and control of user input [23,24].

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:

$$UI = \langle L, C, I, P, S \rangle, \tag{3}$$

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 KB properties,

$S = \{S_1, S_2, \ldots, S_n\}$ is a set of KB states.

The following function is used to build a GUI based on the 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 KB entities represented by definition (1) within the i-th context;

$\{L^{UI}, C^{UI}, I^{UI}, P^{UI}, S^{UI}\}$ is a set of GUI entities represented by the definition (3).

Thus, the contents of the KB are mapped to a set of GUI components. This mapping 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 Experiments

A set of experiments on constructing the ontology of the PrA of the local area network (LAN) state estimation in the process of simulated traffic increase was carried out. The ontology was constructed with the use of the TP. The TP also gives the ability of the inference of expert recommendations on solving the problem situations that occurred in the process of LAN functioning.

The following actions were taken:

1. The group of experts developed FuzzyOWL-ontology and the set of SWRL rules. Ontology and the set of SWRL rules were developed with the use of tools of the TP. Each expert worked with his ontology context. Weight was appropriated to each context. The weight of the expert is defined in the subareas of the PrA. The weight of the expert expressed in value from [0,1].
2. Thirty problem situations of decrease of the LAN performance were simulated on the tested LAN. Technical characteristics of the tested LAN outlined in the table of the results of the experiments. The reasons for the problem situation occurred should be detected.
3. The equipment performances in each simulated situation and the characteristics of the network were put into the KB. The LAN architecture also was put into the KB. The platform inference subsystem performed the output of one or several recommendations for the LAN state correction.
4. A group of expert estimated the correctness of the obtained recommendations. These experts were not involved in the KB development process.

The developed ontology of the LAN state analysis has a hierarchical organization. The developed ontology includes 81 classes, 104 properties, about 200 ontology instances and the set of SWRL-rules.

Table 2 contains the LAN architecture. Also, that table contains some modeled problems in the process of artificial traffic increase in a network. Recommendations on solving the problems generated by the platform tools in the process of inference are presented in Table 2.

Lan architecture is Star network, the number of workstations is 14, the server OS is Windows Server 2008 R2, the network switch is D-Link DGS-3420-28SC, 20 ports, the switch RAM is 2 Mb, the operating mode – 100 Mb/s Full Duplex, the media access control method is CSMA/CD, QoS is on, patch cable type is UTP, the category is 5E, port RJ-45.

5.1 Comparison of the Inference by Fuzzy and Crisp Ontologies

A set of experiments was carried in the TP to compare the interaction of ontological analysis and inference mechanisms by fuzzy and crisp ontologies. The possible problem situations occurred in LAN functioning in the process of traffic increase were simulated.

The LAN of IT-department of Ulyanovsk State Technical University was chosen as an object for the experiments. The comparative analysis of the inference methodologies in the process of modeling problem situations in the LAN functioning is shown on Fig. 5.

Table 2. The modeled processes of artificial traffic increase in a network

Modeled problems	The recommendations
The fast increase of the background load, increase of the user software reaction time	The reason is the overload of the communication channel. It requires network architecture changes. The number of stations in the overloaded domains should be reduced. The stations that create the most significant load should be connected to the switch ports
Half Duplex mode is enabled, the background load increase	The reason is the mismatch between the characteristics of a network switch and the traffic volume. It requires replacement of a switch or change of its settings
Damage of one network patch cable; increase of collision number	The reason is the problem with connection patch cables. The reason is the wrong organization of ground of the computers connected in the local network
Defects of switch ports, increase of CRC errors number, the background load increase	The reason is the problem of the network switch ports or the patch cable damage leading to a problem host
Defects of an Ethernet adapter, the rapid increase of some local and remote collisions	The reason is the problem in the Ethernet adapter settings of the domain server. The user should check the Ethernet adapter and the correctness of its settings

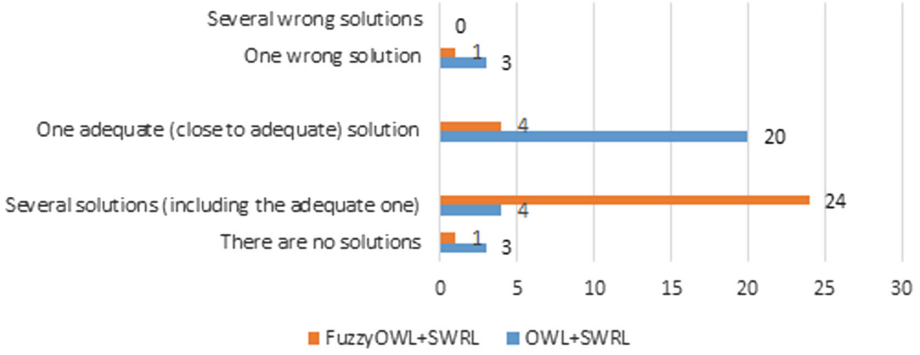


Fig. 5. The results of 30 experiments to the comparison of the inference by fuzzy and crisp ontologies.

The results of experiments show that fuzzy ontology based on FuzzyOWL format showed a more wide range of variants. The variants are sorted by the degree of relevance of recommendations. The risks of loss of possible variants of the logical inference block are significantly reduced.

6 Conclusion

Development of a technological platform for KD is necessary for automation of the process of obtaining essential knowledge about PrA in the internal KB. TP should not require users to have additional skills in knowledge engineering and ontological analysis. TP should provide inference functions.

Contexts of the KB represent the ontology content in space and time. Space contexts allow solving the problem of the expert level of competence in the specific part of the PrA. Each context is associated with a value from 0 to 1 defining the expert level of competence in the part of the PrA.

Time contexts allow using versioning of the PrA ontology. Time contexts give an opportunity to monitor the dynamics of the ontology development and to return to the defined state of the ontology.

The integration of the inference mechanism and the fuzzy ontology in the context of the KB provide a TP of expert decision support for specialists of different organizations. The contents of the KB are mapped to a set of GUI components. Dynamically generated GUI 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.

Fuzzy ontology based on FuzzyOWL format showed a more wide range of variants. The variants are sorted by the degree of relevance of recommendations. The risks of loss of possible variants of the logical inference block are significantly reduced.

References

1. Rubiolo, M., Caliusco, M.L., Stegmayer, G., Coronel, M., Gareli Fabrizi, M.: Knowledge discovery through ontology matching: an approach based on an artificial neural network model. Inf. Sci. **194**, 107–119 (2012)
2. Renu, R.S., Mocko, G., Koneru, A.: Use of big data and knowledge discovery to create data backbones for decision support systems. Procedia Comput. Sci. **20**, 446–453 (2013)
3. Ltifi, H., Kolski, C., Ben Ayed, M., Alimi, A.M.: A human-centred design approach for developing dynamic decision support system based on knowledge discovery in databases. J. Decis. Syst. **22**, 69–96 (2013)
4. Rajpathak, D., Chougule, R., Bandyopadhyay, P.: A domain-specific decision support system for knowledge discovery using association and text mining. Knowl. Inf. Syst. **31**, 405–432 (2012)
5. Bobillo, F., Straccia, U.: FuzzyDL: an expressive fuzzy description logic reasoner. In: Proceedings of the 17th IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2008), pp. 923–930 (2008)
6. Bobillo, F., Straccia, U.: Representing fuzzy ontologies in OWL 2. In: Proceedings of the 19th IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2010), pp. 2695–2700 (2010)
7. Gao, M., Liu, C.: Extending OWL by fuzzy description logic. In: Proceedings of the 17th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2005), pp. 562–567 (2005)

8. Bianchini, D., De Antonellis, V., Pernici, B., Plebani, P.: Ontology-based methodology for E-service discovery. Inf. Syst. **31**, 361–380 (2005)
9. Guarino, N., Musen, M.A.: Ten years of applied ontology. Appl. Ontol. **10**, 169–170 (2015)
10. Guizzardi, G., Guarino, N., Almeida, J.P.A.: Ontological considerations about the representation of events and endurants in business models. In: La Rosa, M., Loos, P., Pastor, O. (eds.) BPM 2016. LNCS, vol. 9850, pp. 20–36. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-45348-4_2
11. Falbo, R.A., Quirino, G.K., Nardi, J.C., Barcellos, M.P., Guizzardi, G., Guarino, N.: An ontology pattern language for service modeling. In: Proceedings of the 31st Annual ACM Symposium on Applied Computing, pp. 321–326 (2016)
12. Hotho, A., Staab, S., Stumme, G.: Ontologies improve text document clustering data mining. In: ICDM 2003, pp. 541–544 (2003)
13. Gruber, T.: Ontology. http://tomgruber.org/writing/ontology-in-encyclopedia-of-dbs.pdf. Accessed 14 Jan 2019
14. Medche, A., Staab, S.: Ontology learning for the Semantic Web. https://www.csee.umbc.edu/courses/771/papers/ieeeIntelligentSystems/ontologyLearning.pdf. Accessed 14 Jan 2019
15. Neo4j: Official site. https://neo4j.com/product. Accessed 14 Jan 2019
16. Dentler, K., Cornet, R., ten Teije, A., de Keizer, N.: Comparison of reasoners for large ontologies in the OWL 2 EL profile. Seman. Web **2**, 71–87 (2011)
17. Pellet Framework Available. https://github.com/stardog-union/pellet. Accessed 14 Jan 2019
18. SWRL: A semantic web rule language combining OWL and RuleML. https://www.w3.org/Submission/SWRL. Accessed 14 Jan 2019
19. Suchanek, F.M., Kasneci, G., Weikum, G.: YAGO: a core of semantic knowledge unifying WordNet and Wikipedia. In: Proceedings of the 16th International Conference on World Wide Web, pp. 697–706 (2007)
20. Yarushkina, N., Filippov, A., Moshkin, V.: Development of the unified technological platform for constructing the domain knowledge base through the context analysis. In: Kravets, A., Shcherbakov, M., Kultsova, M., Groumpos, P. (eds.) CIT&DS 2017. Communications in Computer and Information Science, vol. 754, pp. 62–72. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-65551-2_5
21. Yarushkina, N.G., Filippov, A.A., Moshkin, V.S., Filippova, L.I.: Application of the fuzzy knowledge base in the construction of expert systems. Inf. Technol. Ind. **6**, 32–37 (2018)
22. Shestakov, V.K.: Development and maintenance of information systems based on ontology and Wiki-technology, Electronic Libraries: Advanced Methods and Technologies, Digital Collections, pp. 299–306 (2011). (in Russian)
23. Hattori, S., Takama, Y.: Recommender system employing personal-value-based user model. J. Adv. Comput. Intell. Intell. Inform. (JACIII) **18**, 157–165 (2014)
24. Ruy, F.B., Reginato, C.C., Santos, V.A., Falbo, R.A., Guizzardi, G.: Ontology engineering by combining ontology patterns. In: Johannesson, P., Lee, M.L., Liddle, S.W., Opdahl, A.L., López, Ó.P. (eds.) ER 2015. LNCS, vol. 9381, pp. 173–186. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-25264-3_13