Development of a Fuzzy Knowledge Base as a Core of the Athene Platform

Aleksey Filippov
Department of Information Systems
Ulyanovsk State Technical University
Ulyanovsk, Russia
al.filippov@ulstu.ru

Vadim Moshkin
Department of Information Systems
Ulyanovsk State Technical University
Ulyanovsk, Russia
postforvadim@yandex.ru

Nadezda Yarushkina
Department of Information Systems
Ulyanovsk State Technical University
Ulyanovsk, Russia
jng@ulstu.ru

***Abstract*—The article describes the process of developing a fuzzy knowledge base (KB) as a core of the Athene platform. The content of fuzzy KB is formed as a result of the analysis of the contexts of the problem area (PrA). In this case, the context is a certain "point of view" on the PrA and its features. A graph database (DB) is used as a basis for storing the contents of the KB in the form of an applied 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.**

***Keywords*—ontology, fuzzy knowledge base, context analysis, problem area, graph database, inference, dynamic ui**

I. 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 [1], [2].

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 [3]–[6].

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 fuzzy knowledge base (KB) as a core of the Athene platform [6]. The content of the fuzzy KB is an applied ontology. The basic requirements for fuzzy KB are (fig. 1):

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

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

1) Ontology store:
    - Neo4j;
    - content management module;
    - ontology import/export module (RDF, OWL).
2) Inference subsystem:
    - inference module.
3) A subsystem for interaction with users:
    - screen forms generation module.
4) A subsystem for importing data from external information resources:
    - a module for importing data from external wiki-resources;
    - a module for filling external wiki-resources.

II. The organization of the ontology store of fuzzy knowledge base

Ontology is a model of the representation of the PrA in the form of a semantic graph [7], [8].

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

1) Having a free community version.
2) Native format for data storage.
3) One copy of Neo4j can work with graphs containing billions of nodes and relationships.
4) The presence of a graph-oriented query language Cypher.
5) 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 ontology is some state of ontology, obtained during versioning or building an ontology using different "points of view".

Formally, the ontology can be represented by the following equation:

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

where

t is a number of the ontology contexts;

$T = \{T_1, T_2, \ldots, T_n\}$ is a set of ontology contexts;

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

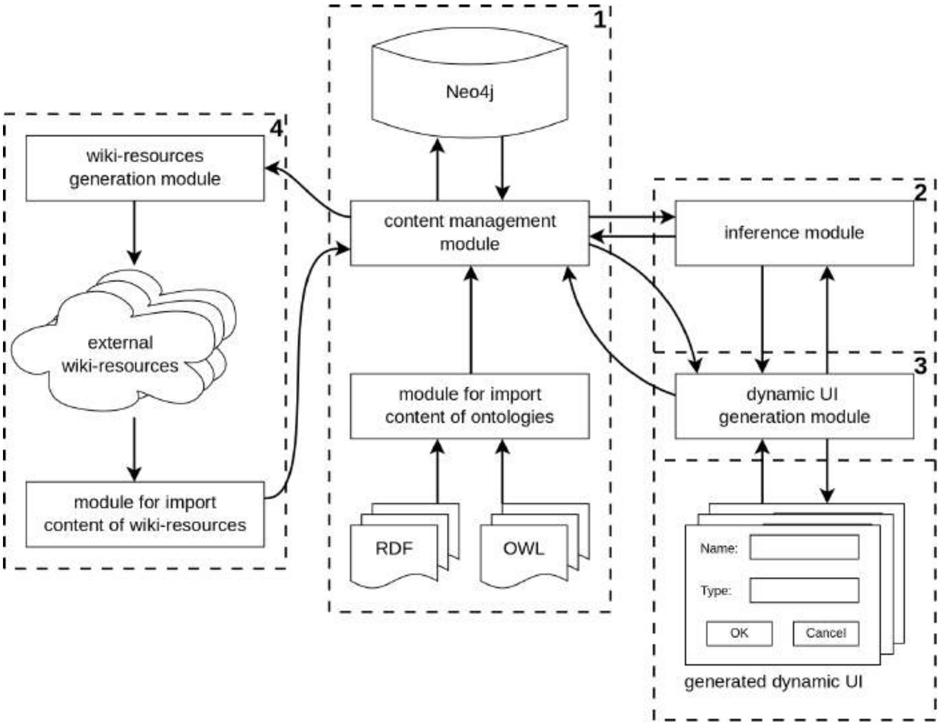

Figure 1. Architecture of a fuzzy knowledge base of the Athene platform.

$C^{T_i} = \{C_1^{T_i}, C_2^{T_i}, \ldots, C_n^{T_i}\}$ is a set of ontology 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 ontology 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 ontology 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 ontology 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 ontology within the i-th context;

R^{T_i} is a set of ontology 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 ontology classes within the i-th context;

$R_I^{T_i}$ is a set of relations defining the 'class-object' ontology tie within the i-th context;

$R_P^{T_i}$ is a set of relations defining the 'class-class property' ontology tie within the i-th context;

$R_S^{T_i}$ is a set of relations defining the 'object-object state' ontology tie within the i-th context;

$R_F^{T_i}$ is a set of relations generated on the basis of logical ontology rules in the context of i-th context.

Principles similar to the paradigm of object-oriented programming are at the basis of the ontology of the fuzzy knowledge base:

- ontology classes are concepts of the PrA;
- classes can have properties, the child-class inherits properties of the parent class;
- objects of ontology 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 fuzzy KB.

III. THE INFERENCE ON THE CONTENTS OF FUZZY KNOWLEDGE BASE

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 [9], [10]. The most popular at the moment reasoners are:

- 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 fuzzy KB.

Currently the Semantic Web Rule Language (SWRL) is used to record logical rules. These SWRL rules describe the conditions under which object a has "nephew-uncle" relation with object c. Formally the logical rule of the ontology of the fuzzy knowledge base is:

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

where

T_i is i-th context of the ontology of the fuzzy KB;

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

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

A^{Cypher} is a Cypher representation of a 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

$Ant = Ant_1 \Theta Ant_2 \Theta \ldots Ant_n$ 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 a consequent (consequence) of a logical rule F^{T_i}.

Figure 2 shows an example of a tree-like representation of a logical rule. This rule describes the nephew-uncle relationship.

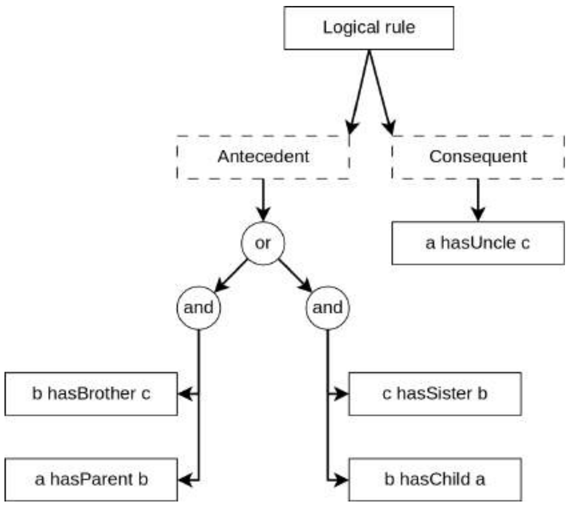

Figure 2. Example of a tree-like representation of a logical rule.

The tree-like logical rule (fig. 2) is translated into the following SWRL:

hasParent(?a, ?b) $\wedge$ *hasBrother(?b, ?c)* $\rightarrow$ *hasUncle(?a, ?c)*
hasChild(?b, ?a) $\wedge$ *hasBrother(?b, ?c)* $\rightarrow$ *hasUncle(?a, ?c).*

and the following Cypher view:

MATCH (a:Object)<-[:RANGE]
-(s:Statementname:"hasParent")
-[:DOMAIN]->(b:Object)
MATCH (b:Object)<-[:RANGE]
-(s1:Statementname:"hasBrother")
-[:DOMAIN]->(c:Object)
MERGE (a)<-[:RANGE]
-(s2:Statementname:"hasUncle")
-[:DOMAIN]->(c)
MATCH (b:Object)<-[:RANGE]
-(s:Statementname:"hasChild")
-[:DOMAIN]->(a:Object)
MATCH (c:Object)<-[:RANGE]
-(s1:Statementname:"hasSister")
-[:DOMAIN]->(b:Object)
MERGE (a)<-[:RANGE]
-(s2:Statementname:"hasUncle")
-[:DOMAIN]->(c).

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 ontology of the fuzzy KB. These relations correspond to the antecedent atoms of the logical rule. Formed relationships provide the inference from the contents of the fuzzy KB.

IV. Building a Graphical User Interface (GUI) based on the contents of a fuzzy knowledge base

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

You need to map the fuzzy KB ontology entities to the GUI elements to build a GUI based on the contents of the fuzzy 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 ontology classes;

$I = \{I_1, I_2, \ldots, I_n\}$ is a set of ontology objects;

$P = \{P_1, P_2, \ldots, P_n\}$ is a set of properties of ontology classes;

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

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

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

where

$\{C^{T_i}, I^{T_i}, P^{T_i}, S^{T_i}, F^{T_i}, R^{T_i}\}$ is a set of ontology entities of fuzzy KB represented by expression 1 within the i-th context;

$\{L, C, I, P, S\}$ is a set of GUI entities of fuzzy KB represented by the expression 2.

Thus, the contents of the fuzzy KB are mapped to many 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.

V. Interaction of fuzzy knowledge base with external wiki-resources

At present, wiki-technologies are used to organize corporate KB. It is necessary to solve the task of importing the content of such wiki-resources into fuzzy KB [12], [13]. Table I contains the result of mapping the fuzzy KB ontology entities to the wiki resource entities.

Thus, it becomes possible to import the content of external wiki resources for initial filling of KB contents. There is also the possibility of the reverse process – generation of wiki-resources based on the contents of fuzzy KB.

Table I
Matching ontology elements of fuzzy KB and wiki-resources

The ontology element of fuzzy KB	The element of wiki-resources
Class	Category
Subclass	Subcategory
Object	Page
Class properties	The infobox elements (properties)
Object states	The infobox elements (values)
Relations	Hyperlinks

VI. Conclusion

Thus, the use of fuzzy 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 fuzzy KB by translating SWRL-rules into Cypher-structures;
- building a graphical user interface based on the contents of fuzzy KB;
- automated import of knowledge from internal and external wiki-resources.

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.

References

[1] Bova V.V., Kureychik V.V., Nuzhnov Ye.V. Problemy predstavleniya znaniy v integrirovannykh sistemakh podderzhki upravlencheskikh resheniy [Problems of representation of knowledge in integrated systems of support of administrative decisions]. Izvestiya YUFU [News of SFU], 2010, vol. 108, no. 7, pp. 107-113.

[2] Chernyakhovskaya L.R., Fedorova N.I., Nizamutdinova R.I. Intellektual'naya podderzhka prinyatiya resheniy v operativnom upravlenii delovymi protsessami predpriyatiya [Intellectual decision support in the operational management of business processes of the enterprise]. Vestnik UGATU. Upravleniye, vychislitel'naya tekhnika i informatika [Bulletin of the USATU. Management, Computer Science and Informatics], 2011, vol. 42, no. 2, pp. 172-176.

[3] Zagorul'ko YU.A. Postroyeniye portalov nauchnykh znaniy na osnove ontologii [Building portals of scientific knowledge on the basis of ontology]. Vychislitel'nyye tekhnologii [Computing technologies], 2007, vol. 12, no S2, pp. 169-177.

[4] Karabach A.Ye. Sistemy integratsii informatsii na osnove semanticheskikh tekhnologiy [Information integration systems based on semantic technologies]. Nauka, tekhnika i obrazovaniye [Science, technology and education], 2014, vol. 2, no. 2, pp. 58-62.

[5] Tuzovskiy A.F. Razrabotka sistemy upravleniya znaniyami na osnove yedinoy ontologicheskoy bazy znaniy [Development of knowledge management systems based on a single ontological knowledge base]. Izvestiya Tomskogo politekhnicheskogo universiteta [Proceedings of Tomsk Polytechnic University], 2007, vol. 310, no. 2, pp. 182-185.

[6] Filippov A.A., Moshkin V.S., Shalayev D.O., Yarushkina N.G. Yedinaya ontologicheskaya platforma intellektual'nogo analiza dannykh [Unified Ontological Data Mining Platform]. Otkrytye semanticheskie tekhnologii proektirovaniya intellektual'nykh system [Open semantic technologies for intelligent systems], 2016, pp. 77-83.

[7] Moshkin V.S., Yarushkina N.G. Metodiki postroyeniya nechetkikh ontologiy slozhnykh predmetnykh oblastey [Methods for constructing fuzzy ontologies of complex subject domains]. Otkrytye semanticheskie tekhnologii proektirovaniya intellektual'nykh system [Open semantic technologies for intelligent systems], 2015, pp. 401-406.

[8] Tarasov V.B., Kalutskaya A.P., Svyatkina M.N. Granulyarnyye, nechetkiye i lingvisticheskiye ontologii dlya obespecheniya vzaimoponimaniya mezhdu kognitivnymi agentami [Granular, fuzzy and linguistic ontologies to provide mutual understanding between cognitive agents]. Otkrytye semanticheskie tekhnologii proektirovaniya intellektual'nykh system [Open semantic technologies for intelligent systems], 2012, pp. 267-278.

[9] Makhortov S.D. Ob algebraicheskoy modeli raspredelennoy produktivnoy sistemy [On the Algebraic Model of a Distributed Production System]. KII-2016. 2016, vol. 1, pp. 64-72.

[10] Moshkin V.S., Yarushkina N.G. Logicheskiy vyvod na osnove nechetkikh ontologiy [Logical output based on fuzzy ontologies]. Integrirovannyye modeli i myagkiye vychisleniya v iskusstvennom intellekte [Integrated models and soft computations in artificial intelligence], 2015. vol. 1, pp. 259-267.

[11] Gribova V.V., Kleshchev A.S. Upravlenie proektirovaniem i realizatsiei pol'zovatel'skogo interfeisa na osnove ontologii [Managing the design and implementation of the user interface based on ontologies]. Problemy upravleniya [Management issues], 2006, vol 2, pp. 58-62.

[12] Shestakov V.K. Razrabotka i soprovozhdeniye informatsionnykh sistem, baziruyushchikhsya na ontologii i Wiki-tekhnologii [Development and maintenance of information systems based on ontology and Wiki-technology]. Elektronnyye biblioteki: perspektivnyye metody i tekhnologii, elektronnyye kollektsii [Electronic Libraries: Advanced Methods and Technologies, Digital Collections], 2011, pp. 299-306.

[13] 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, 2007, pp. 697-706.

РАЗРАБОТКА ЯДРА ПЛАТФОРМЫ Athene – НЕЧЕТКОЙ БАЗЫ ЗНАНИЙ

А. Филиппов, В. Мошкин, Н. Ярушкина
Кафедра «Информационные системы»
Ульяновский государственный технический университет,
Россия, Ульяновск

В работе описывается процесс разработки ядра платформы Athene – нечеткой базы знаний (БЗ), содержимое которой формируется в результате анализа контекстов проблемной области (ПрО). В данном случае под контекстом понимается некоторая «точка зрения» на ПрО и ее особенности. В качестве основы для хранения содержимого БЗ в виде прикладной онтологии используется графовая база данных (БД). Представлена попытка реализовать механизм логического вывода по содержимому графовой БД. Для упрощения работы с БЗ используется механизм динамической генерации экранных форм интерфейса пользователя.