Applied Computer Science, vol. 11, no. 4, pp. 70–82

Submitted: 2015-09-20
Revised: 2015-10-30
Accepted: 2015-12-03

ontology, machine learning, logic predicate

Oksana OBORSKA[*], *Dmytro MAHEROVSKYJ*[**],
Roman VOVNJANKA[***]

COMPUTER SYSTEM FOR AUTOMATED ONTOLOGY BUILDING BASIC CROCUS

Abstract

The article exposes the approach developing a computer system of automated ontology building based on creation of architecture system ontology synthesis CROCUS (Cognition Relations or Concepts Using Semantics) software model. The basic modules of the system and its operations are described. The choice of software tools for implementation was described. Example of SDK decision for system realization was substantiated. The using of this system allows filling the domain ontology in automatic mode.

1. INTRODUCTION

The analysis of researches and develops in the branch of intellectual information systems and Internet services gives a ground to suggest that such soft/hardware decisions:

- realization of the ontology synthesis system as a subsystem of the Internet portal system [1];
- using OWL as a knowledge presentation language;

[*] Lviv Polytechnic National University, Ukraine, 79013, Lviv, Bandera str., 28a, oksana949@gmail.com

[**] Lviv Polytechnic National University, Ukraine, 79013, Lviv, Bandera str., 28a, maherovskyi@gmail.com

[***] Lviv Polytechnic National University, Ukraine, 79013, Lviv, Bandera str., 28a, vovnianka@ukr.net

- using HTN and OWL-S as structures of the automated knowledge base planning language;
- Java API for Protégé OWL as the API and the library processing classes, in particular for the machinery studying (reinforcement learning) of the OWL-ontology and knowledge bases;
- Link Grammar Parcer as an instrument of the grammatically-semantic analysis of English text documents;
- Apache-PHP-MySQL as a software tool to build a web-portal based user interface;
- we get as a web service for automatized access to search engines with a query, formed from the keywords;
- SWRL as a logical new knowledge output language with deductive and inductive methods;
- WordNet as the basic glossary of English.

An ontology in OWL language contains a high-level meaning automation of the subject area. A high-level ontology provides:

- a logical output of new knowledge with the addition of new messages with the context;
- the verification of the validity of obtained statements;
- the evaluation of the probability of the message sources;
- the ensuring of the knowledge base logical integrity.

Extensive researches are conducted in the field of information technologies, especially artificial intelligence. The advances are strengthened by the creation of wide range of software and universal and specialized computer equipment that uses it. All this is accompanied by the rapid and intensive formation of the new scientific terminology that is not always commonly acknowledged. Therefore to avoid ambiguity we should give the basic definitions of concepts used in this paper.

The notion of 'knowledge' remains most common and least clearly defined yet. In this paper, we assume that knowledge is useful information. Unlike the information that is measured as media volume needed for its storage, knowledge should be measured as a benefit of using corresponding information. Knowledge acquisition can in principle take place only when and where there is its (potential) carrier – an intellectual agent which has information about its current and desirable state as a goal, achievement problem and motivation, and a strategy (plan) to achieve its goals. Only then the information can be used to solve these problems and thus serve as knowledge to their carrier – an intelligent agent.

We are considering ontology as formal explicit representation of common terminology and its logical interdependence for a certain subject domain. An ontology formalizes an intensional of the domain – e.g. a set of rules in terms of formal logic, while its extensional is defined in the knowledge base as a set of facts about instances of concepts and relationships between them. The process of filling the knowledge base called knowledge markup (further – KM), or ontology population (further – OP), methods and tools for automatic (semi-automatic) ontology structure development – ontology learning (OL). OL methods in turn are based on the methods of natural language processing (NLP) [2] and machine learning (ML). Far less attention is paid to the approaches developed in the field of automated planning (AP). Knowledge acquisition (KA) in particular from text documents using the NLP methods is perhaps the only way for automatic construction of ontology, which however cannot replace an OL as a scientific discipline, remaining its key instrument. A role of ontology makes a fundamental difference between OL and KA because for OL it is not only the tool but also a target and a performance criterion for methods and tools developed during OL researches.

The method of recognizing the logic content of the natural text document i.e. natural language understanding (NLU) is based on an information technology of semantic text analysis, i.e. text mining (TM), which in turn can be approximately determined as the process of identification information that can be useful for solving certain tasks [3]. On the other hand, the research area TM includes NLU in the part where it acts as a scientific discipline – linguistic tool for translation natural language (NL) documents to formal knowledge representation languages for further formal analysis. NLU in turn considered as a section of NLP [4] (Fig. 1).

Currently TM is an actively developing scientific discipline. It has not well developed institutions (no universally recognized textbooks, lectures, chairs on this subject) and therefore is in a somewhat uncertain status. Some experts interpreted TM too broadly and include in it all IT techniques which deals with NL texts, others understood TM too narrowly as a particular case of statistical data analysis – data mining (DM), yet others tend to believe this discipline extension or replacement of the information retrieval (IR) discipline. In most cases, a statement of the problem depends on the subject domain in which it is formulated. TM includes very different approaches to this problem by a level of analysis – a simple classification using a syntactic parser and identification the specific values of the semantic structures as provided, for example, in a quite advanced project GATE, up to complex predictive analysis, aimed at finding a solution of the problem. It is the second approach is considered in this paper – an intelligent content recognition (understanding) NL text as a main source of knowledge today.

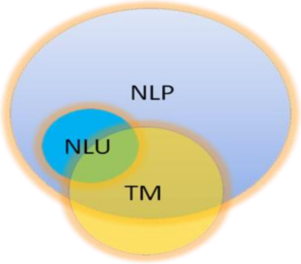


Fig. 1. Interference of scientific disciplines in natural text processing – TM includes NLU in field of targeted logical analysis [source: own study]

1.1. Java API Protégé-OWL

The machinery education is provided with the means of Java API Protégé-OWL. These means contain libraries of classes, which realize methods to work with OWL-structures like reading and addition. Therefore, the means of machine learning are working in addition to the OWL-ontology. It takes templates of grammatically semantic structures to recognize statements (the first order predicates) into the research and/or educated texts including new elements as the result of such recognition. Link Grammar Parcer divides a grammatically correct approving sentence into interconnected pairs of words. LGB contains a table that has all the conformities between grammar constructions of English and syntax-semantically links between words (intellections). LGP API allows to link this table to OWL-ontology, so the table can adapt in the process of learning the given object area dynamically.

Java API Protege-OWL [5] based means of machinery education contain a generalized description of the semantic link, which serves as the template for generating new types of semantic link during studying. In addition, it forms appropriate vectors and indications of these links to form and identify semantic links in a text. Herewith, properly classes of links and their properties adding to the OBP. Exemplars of those classes are for the description of existing and new classes of an ontology by their use as first rank predicates.

1.2. Ontology learning

The process of ontology learning [6] is widely discussed lately. But all elaborated approaches do not take into account the problem of reaching an optimal structure and dimension of an ontology. It is known that computational complexity of graph processing algorithms not allow to work (at least in real time scale) with especially large ontologies, for example in case of solving tasks of eliminating knowledge inconsistency. Therefore extremely important not simply add all accessible relevant knowledge data to an ontology but refine existing knowledge structures inside defined optimal volume. This task must follow some criteria of optimality most valuable of which is data usefulness, applicability or, other words, knowledge pertinence.

To estimate data usefulness as the main criteria of optimality it is possible to count a frequency of mention of connected with it concepts and relations in ontology. This is a formal evidence of data importance, because ontology represents sphere of agent interests if and only if learning procedure will be correct enough. Such approach was considered in our previous work.

More precise approach must take into account the main task of agent's activity or, to be more specific, hierarchy of tasks and a gain from its solving as the most adequate criterion of information usefulness. It is possible because an ontology should contain an explicit specification of that task hierarchy. Moreover, if a first dimension of an ontology structure is a taxonomy, second must be built as a hierarchical task network (HTN) (Fig. 2).

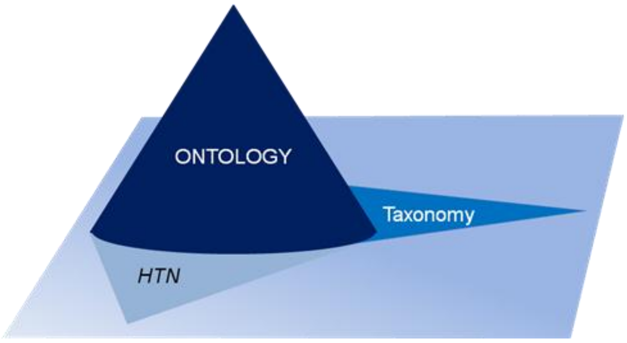


Fig. 2. Two different dimensions an ontology

An ontology has two different dimensions – as a taxonomy of concepts and as a hierarchical task network (HTN) [7].

Thus ontology learning process is based on NLP procedures with aim to find out subtasks of the agent's main task, appropriate task solving strategies and their components – resources, actions (operators), methods, preconditions, effects, dependencies and other restrictions. It is performed on different levels of coverage of text – from separate found term definitions up to the whole message meaning as a problem solving recipe. In any case all new information or supplements an optimal strategy represented in an ontology, or cannot do it and is rejected as unnecessary.

Learning process in details consists of next stages:

- a task from own ontology, which is appropriate to the task described in a message, extracted from NL text document, must be selected;
- new approach taken from message is applied temporarily to solve the selected task;

- a maximum expected utility of a new strategy is estimated and compared to previous one;
- if an utility value growing, the ontology is corrected, otherwise message data is stored in a knowledge base as a reference information about it's source.

Agent also learns different task solving patterns – decision making using different heuristics. In one of them for simplicity each task could be represented as partially observable Markov decision process (POMDP) that is a generalization of the standard completely observable Markov decision process that allows estimate optimal strategy using imperfect (incomplete) information about the state of the system. Such formalism possess well developed solving algorithms therefore is useful for implementation.

ZMN ontologies make sense only as the part some intellectual system. Optimal decisions in our opinion are that, where such an intellectual system [8-10] in the information search system for which an adaptive ontology is an instrument for information research, analysis and classification on the one hand, which uses search instrumentalities to provide data for its filling, new predicates and rules synthesis, learning new means and semantic links on the other. An intellectual system of information searching based on the adaptive ontology, material science knowledge base, a database of scientific publications became such a decision.

A developed architecture of the ontology synthesis system was realized with the usage of selected and descripted means and program-technically decisions as a CROCUS (Cognition Relations Or Concepts Using Semantics) [11].

2. THE PROBLEM FORMULATION

2.1. New knowledge evaluation

All the procedures are needed to evaluate the knowledge contained in the analyzed message. As has been mentioned above it depends on message context which is explicitly expressed in an ontology of the agent and optimal strategy of reaching its goal that it contains. The evaluation method is based on the expected value of perfect information – $EVPI$:

$$EVPI = EV \mid PI - EMV, \tag{1}$$

where: EMV – is the probability weighted sum of possible payoffs for each alternative;

$EMV = \max_i \sum p_j R_{ij}$ (2), $\sum_i p_j R_{ij}$ (3), is the expected payoff for action i ; $EV \mid PI$ – the expected or average return if we a priory have "perfect" (i.e. new) information for best i choice:

$$EV \mid PI = \sum_i p_j (\max R_{ij}) \tag{4}$$

To estimate new knowledge $EVPI$ we must have the value of EMV for each solving approach (action for (3)) to each task from HTN of our ontology and with aim to obtain them all we must create and solve appropriate POMDP task:

$$EMV_i \equiv U(S_i) = R(S_i) + \gamma \cdot \max_{A_{ik}} \sum_k P(S_i, A_{ik}, S_j) U(S_j) \tag{5}$$

Equation (5) describes the reward for taking the action giving the highest expected return. Additional information decreases model uncertainty therefore expected common reward (utility) will be not less than without such information.

Using them for particular domain model it is possible to evaluate expected utility for both cases: with new information and without it.

2.2. Reliability evaluation

All obtained information is verified for its consistency and if any contradictions appear a logical conflict is solving by rejecting data with less reliability of its source. In such case the reliability of the source D also decreases according to appropriate formula:

$$D_{n,i+1} = \frac{D_{n,i}}{(2-s)} + \frac{1-D_{n,i}}{2} \cdot s$$

where: s – the truth of the statement that takes the value 1 if the statement is true or 0 – otherwise, i – step number confirmation/denial of the truth of the one statement of n-th source.

The purpose of this article is to develop a computer system building automated ontology base.

3. MAIN PART

3.1. CROCUS system

The overall concept of CROCUS [12] is introduced at the (Fig. 3). A subsystem of the ontology education uses educational texts of annotations of scientific publications from article DB. The system forms a plural of key words to fill the DB in. It chooses the main metadata about the publications in the defined subject area in Internet (ScienceDirect, CiteSeer, Wiley Online Library, Springer) including their annotations, which become the core of analysis and ontology learning.

The essence of the knowledge extraction method from the natural text document is into building of the intellectual agent activity strategy – an informational model of the recognition subject of its specification based on dedicated from recognized text document data. A plan is considered to be a specific optimal strategy realization of some task, which has an intellectual agent within the subject area.

The plan is built with the same with informational model formal knowledge representation language – a database of an intellectual agent. Considering that, such a knowledge base is already an overall plan of intellectual agent functioning, build basing on the natural text recognition is a sub-plan. It means that it is a specification of an overall plan and it bases on it. A value of information, received as a result of recognition the context of a text document is determined as increasing of the updated intellectual agent functioning plan expected utility.

Scientific publications range for the relevance to the users informational demands, for the conformity to ontology, which displays these demands. An analysis of each annotation as natural text is made, builds its image in the terms of ontology as predicates and rules in this purpose. These predicates and rules are added in the knowledge base of the system and the expected utility of an intellectual agent is calculated again. A system puts those publications nearer to the beginning of the list, which data including leads to the greater reliability change with such a type of ranging.

A system can adopt to the users requirements by saving his preference system in the DB. Each user can perform an education of his ontology. The system saves the data about this process, leads the session statistics, provides the possibility to correct the education errors and does backtracking to previous versions of an ontology.

CROCUS system modules are shown at the (Fig. 3). A client has a possibility to control the priority of document ranging, to correct their order in the list of the most important (relevant to the client informational requirements) document and classify them with the help of graphic interface. The most important documents are used for ontology education and building of the efficient sets of key schemas and new, received from Internet, articles (their metadata including annotation) insert into the DB of publications with the link with preferences of the user and other prerequisites of the document receiving. First of all, its sources.

An annotation processing happens after its previous processing, conversion into the massive of predicates as a result of grammatically-syntax analysis of Link Grammar Parser. Formed annotation models are supplemented with semantically near ontology predicates – the context of this annotation. Supplemented annotation models are compared between themselves to calculate the semantic length between their semantic weight midpoints and so the nearest by content documents are chosen with their further ranging and classification.

3.2. Main functions of CROCUS

An interactive automatic building of the problem area ontology. Searching, saving and classification (ranging) of scientific publications as in interactive semiautomatic as in automatic mode.

Each of these functions is realized with its base set of functionality modules but a part of them was a double appointment. CRONUS is realized as an object oriented paradigm by using Java as a hierarchy of code classes, which copies call each other with determined at that moment parameters or they interact through throw events and/or handlers. Most of them have a Swing graphical interface and AWT libraries. All the connected libraries have an open source status. Project has a full functionality and has all the necessary means for its development (evolution). A functional assignment of the main CROCUS system modules is shown at the (Fig. 3).

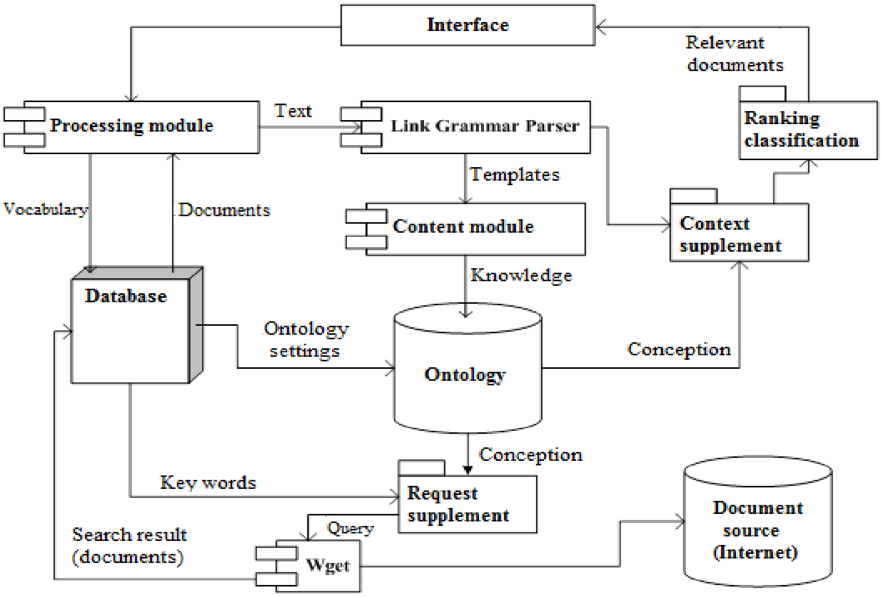
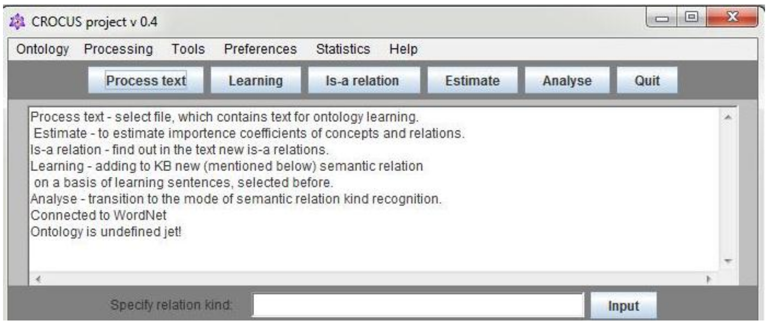


Fig. 3. CROCUS modules [source: own study]

As an implementation of presented KD concept the intellectual information retrieval system CROCUS (Cognition Relations Or Concepts Using Semantics) was created. The system was built on the basis of Java Protégé-OWL API, using OWL-DL as a knowledge representation language. For providing of recognition new do-main concepts at interactive learning mode the WordNet API was included in the system. Through the use of DBMS MySQL it is possible for the CROCUS system to store, process and use during learning process a statistics for many domains and many users simultaneously and independently. The main interface of the CROCUS application is presented at (Fig. 4).



```
Process text - select file, which contains text for ontology learning.
 Estimate - to estimate importence coefficients of concepts and relations.
Is-a relation - find out in the text new is-a relations.
Learning - adding to KB new (mentioned below) semantic relation
 on a basis of learning sentences, selected before.
Analyse - transition to the mode of semantic relation kind recognition.
Connected to WordNet
Ontology is undefined jet!
```

Fig. 4. The main window of CROCUS user interface [source: own study]

It performs two main interdependent functions: learn an ontology in both super-vised and unsupervised modes, and search new knowledge in annotations of scientific articles using ontology with aim to recognize it and estimate measure of its importance.

3.3. Functionality of CROCUS modules

The basic system control element in CROCUS is ControlGUI module (user graphical control interface). This module has a graphical interface by which user can execute procedures, which are provided by the system functionality. Module has the main menu and its main functions are in the toolbar. Control output is carried out at the appropriate text panel. There is an output field and an input field at the underside of the main window to specify the semantic link type at the process of ontology learning using learning sentences (Fig. 4**)**.

Despite the importance of the effective dialog between system and user, a great attention during developing was to the graphical interface design, its intuitiveness and pithiness with a functional completeness of project tasks realization. In addition, there is a possibility to zoom interface to expand its functionality. Despite the intense competition between the similar projects, a great importance was to create a recognizable logotype, which will be replied to the content of the word CROCUS. An experience of such foreign projects confirms an efficiently of such an approach (Protégé to work out with OWL-Ontology, project GATE, etc.). That is why all the windows in dialogue with user are decorated with CROCUS logotype – an illustration of 6-petal saffron (crocus) flower. Professional designers developed an image and the design of main windows interface.

System has an internationalization of whole text dialogues. User can choose a comfortable interface language from four available. There is no problem in language addition. A dialog language file MessagesBundle_xx_XX.properties has to be translated, where XX – the code of a language (RU – Russian, UA – Ukrainian etc.).

To choose the dialogue language you have to choose a subparagraph 'language' into the paragraph of the main menu 'Preferences'.

3.4. Justification of the SDK choose

A using of SDK common libraries gives a chance to avoid unjustified overrun of time, finances and human resources for their redevelopment. Therefore, there is a wide list of investigated currently working analogue projects in this work. Most of them use the concept of open source code and free licensing. The leading developers groups provide their projects with API (Application Programming Interface) means, so the functionality of these objects may be efficiently used by cataloged and well-documented procedures and functions with appropriate settings.

Co-authorship of SDK developments worked out principles of software application usage with different license agreements [12–16]. In addition, they can take part in support and development of existence projects, so each developer has a possibility to get and install these project or libraries and use them as he or she wants. Such internet portals as SourceForge.net contain all the necessary instrumentals for documentation and support projects of every level of difficulty, readiness, access level and popularity between users. Developers actively use special foundation servers, which provide collective (let it be 1000 developers) software developing. The most popular foundation server is Git. It can be installed separately as individual or corporative server and it is possible to use a global GitHub server.

Researches show, that most of the develops in text documents natural processing, almost all the develops in ontology learning are performed on Java. Moreover, Java is dominating between projects languages at SourceForge resource.

A decisive argument to Java usage is an accessibility of Protégé-OWL Java API of Stanford University (USA), because Stanford Center for Biomedical Informatics Research became a flagman of practice developments in OWL SDK-s.

Projects made by Java:

- Gate [http://gate.ac.uk/] – a couple of text documents processing means to find a new knowledge;
- owlapi.sourceforge.net – another one Java project, which is an OWL documents processing Java classes library with broad functionality;
- Pellet [http://clarkparsia.com/pellet/] – a logical output machine to realize thinking (new knowledge output) from OWL 2.0 knowledge base.

4. CONCLUSIONS

Therefore, this work shows an approach to develop an automatized basic ontology building. An architecture of ontology synthesis system as CROCUS (Cognition Relations Or Concepts Using Semantics) software model was created. The main system modules and their appointment were described. A decision of SDK for system realization was substantiated. A usage of such a system can fill an ontology of subject area automatically.

REFERENCES

[1] BIAO QIN, SHAN WANG, XIAOYONG DU, QIMING CHEN, QIUYUE WANG: *Graph-based Query Rewriting for Knowledge Sharing between Peer Ontologies*. Information Sciences, Vol. 178, No. 18, 2008, pp. 3525–3542.

[2] MULLER H. M., KENNY E. E., STERNBERG P. W.: *An Ontology-Based Information Retrieval and Extraction System for Biological Literature.* PLoS Biol., Vol. 2, No. 11, 2004.

[3] DONINI F., NARDI D., ROSATI R.: *Description Logics of Minimal Knowledge and Negation as Failure.* ACM Transactions on Computational Logic, Vol. 3, No. 2, 2002, pp. 177–225.

[4] BOUILLET E., FEBLOWITZ M., LIU Z., RANGANATHAN A., RIABOV A.: *Knowledge Engineering and Planning Framework based on OWL Ontologies.* In ICKEPS-2007.

[5] MEINEL CH., LINCKELS S.: *Semantic interpretation of natural language user input to improve search in multimedia knowledge base*. Information Technologies, Vol. 49, No. 1, 2007, pp. 40–48.

[6] HAUSKRECHT M.: *Value-Function Approximations for Partially Observable Markov Decision Processes*. JAIR., Vol. 13, 2000, pp. 33–94.

[7] EUZENAT J.: *An API for Ontology Alignment*. In Sheila A. McIlraith, Dimitris Plexousakis, and Frank van Harmelen, editors, Proceedings of the 3rd International Semantic Web Conference, Vol. 3298 of LNCS, Berlin, 2004, pp. 698–712.

[8] BONTCHEVA K., SABOU M.: *Learning Ontologies from Software Artifacts: Exploring and Combining Multiple Sources,* Workshop on Semantic Web Enabled Software Engineering (SWESE), Athens, G. A., USA 2006.

[9] STOILOS G., STAMOU G., KOLLIAS S.: *A String Metric For Ontology Alignment*. In: GIL Y., MOTTA E., BENJAMINS V. R., MUSEN M. A. (editors): Proceedings of the 4rd International Semantic Web Conference (ISWC). Vol. 3729 of LNCS, Berlin, 2005, pp. 624–637.

[10] DOSYN D., LYTVYN V., NIKOLSKY Y., PASICHNYK V.: *Intelektualjni systemy, bazovani na ontolijah [Intelligent system based on ontology]*. Cyvilizacija, Lviv, Ukraine, 2009, p. 414.

[11] WONG W., LIU W., BENNAMOUN M.: *Ontology learning from text: A look back and into the future.* ACM Computing Surveys (CSUR), Vol. 44, No. 4, 2012, p. 20.

[12] QIU J., HAASE P., GUILIN Q.: *Combination of Similarity Measures in Ontology Matching using the OWA Operator*. Proceedings of the 12th International Conference on Information Processing and Management of Uncertainty in Knowledge-Base Systems (IPMU'08), 2008.

[13] KNAPPE R., BULSKOV H., ANDREASEN T.: *Perspectives on Ontology-based Querying.* International Journal of Intelligent Systems, 2004, http://akira.ruc.dk/~knappe/publications/ijis2004.pdf.

[14] HATZI O., VRAKAS D., BASSILIADES N., ANAGNOSTOPOULOS D., VLAHAVAS I.: *The PORSCE II Framework: Using AI Planning for Automated Semantic Web Service Composition.* The Knowledge Engineering Review, Cambridge University Press, Vol. 02:3, 2010, pp. 1–24.

[15] LYTVYN V., MEDYKOVSKYJ M., SHAKHOVSKA N., DOSYN D.: *Intelligent Agent on the Basis of Adaptive Ontologies.* Journal of Applied Computer Science, Vol. 20, No. 2, 2012, pp. 71–77.

[16] JACSO P.: *The impact of Eugene Garfield through the prizm of Web of Science.* Annals of Library and Information Studies, Vol. 57, 2010, p. 222.