Ontology Evaluation Based on the Visualization Methods, Context and Summaries

Kristína Machová, Jozef Vrana, Marián Mach, Peter Sinčák

Department of Cybernetics and AI, Technical University, Letná 9, 04200 Košice, Slovakia, {kristina.machova, jozef.vrana, marian.mach, peter.sincak}@tuke.sk

Abstract: The paper focuses on the field of ontology evaluation and visualization. Ontologies represent the essential technology for the development of the Semantic web applications. This technology has been proven to be useful in a range of applications for data manipulation and administration. The paper introduces an ontology visualization approach based on descriptive vectors. It offers the design of descriptive vectors representation for an ontology domain and also the algorithm design for generation of the descriptive vectors. This approach offers quick overview of the given ontologies content. In addition, this work presents the design of methods for comparison and evaluation of various ontologies based on descriptive vectors. Moreover, it introduces a method for ontology placing in the context within an ontological space (the map). Finally, a method for administration of user navigation in the ontological space is presented.

Keywords: the Semantic web, ontology evaluation, ontology visualization, descriptive vector, key concept, user navigation

1 Introduction

The paper is associated with the Semantic web, which has become an inseparable part of our life. The current web is heterogeneous – it contains millions of information sources with various structures. Web users have sometimes big problems to process the enormous amount of information available on the Internet. The base of the web is represented by the technology of hypertext references, which enable interconnection of one source to another one. In such net, the user orientation is difficult. In this case, the semantics is presented in the web, but it is presented only in an elementary form of hypertext links. The links can be considered as instances of some form of relations, but without any exact semantic specification. They contain a lot of accessible information chunks, which are readable by people but unreadable by machines. Nowadays, many researchers try to find a way how to process the web information automatically by machines

within the Semantic web searching. The Semantic web development brought up the increased interest about the technologies used in the process of creation and using of the Semantic web applications (for example XML, RDFS for metadata and RDF, OWL for ontologies). The Semantic web assumes a new knowledge deduction from the knowledge explicitly presented in some given ontology, for which logics – another the Semantic web technology - can be used.

The Semantic web development stimulates the increased interest in the ontologies, in the developing standards and creating ontologies regarding these standards. This effort has resulted in a great number of ontologies available on the Internet these days, sustaining the premise that ontologies are useful in many areas: digital libraries, management of information, the Semantic web or knowledge based systems. For example, in the paper [22] an approach towards modelling a classical expert system using an ontology-based solution is presented. The increasing amount of accessible ontologies leads to a need of methods for effective visualization of the ontology structure in order to support ontology management and searching. The [11] presents an application of ontologies and semantic technologies to the creation of an enhanced management system. The [5] presents a tool-based semantic framework that uses ontology and requirements boilerplates to facilitate the formulation and specification of security requirements.

The increased interest in the ontologies evokes a need of their processing, indexation, searching and reusing. Within the last few years, some platforms for ontology searching like Swoogle [6] and Watson [3] were developed. The work [10] uses the Semantic web technologies to enable content representation to be independent of particular content presentation platforms. But no tool supports decision making connected to the question, which ontology is the best one from the point of view of a user and his/her domain of interest. The decision making and appropriate ontology searching can be made very complex by means of the keywords ambiguity and lack of explicit data about the domain that the ontology covers. The paper [17] introduces a domain specific language called SWSM for a model-driven development of web services. Also, the language of the ontology has to be taken into account. The work [1] proposes a semi-automatic procedure to create ontologies for different natural languages.

In this work we argue to develop an easy way enabling to obtain a general impression of what a particular ontology is about. The mentioned decision process can be supported by ontology visualization tools for the inspection of all the concepts of an ontology and their relations. This is the reason for considering ontology visualization. The aim of the paper is to propose a new tool for ontology visualization using descriptive vectors – the tool represents a novel approach to providing a quick overview of the content of given ontologies without necessity for long searching and exploration of the ontologies. This approach is an alternative to FCA (Formal Concept Analysis) [25], but our approach is more effective, because it is not so exhaustive and complex. The descriptive vector of an investigated ontology and the descriptive vector of the domain of user interests

are used for user navigation within the process of searching and selection of an appropriate ontology. This work presents a design of methods for comparison and evaluation of different ontologies based on descriptive vectors. Within the design process, the ontology visualization and evaluation are not considered separately. The ontology visualization and evaluation create design environment simultaneously serving for all needed functions in one integrated access.

2   Ontology Visualization

Because ontologies can reach extreme size and complexity, the developing of ontology visualization tools is necessary. Ontology visualization methods according to [13] can be divided into the following groups: *Indented lists, Graph and tree structures, Zoom-able techniques, Space-filling techniques, Focus + context or distortion access, 3D information landscapes*.

The *indented lists* form a simple and intuitively understandable group of visualization methods, which presents classes as nodes in indented collapsible tree. System Protégé [12] serves such representation. A disadvantage of it lies in lucidity connected with large number of classes and higher number of nested levels.

The *graph and tree structures* represent a very suitable metaphor for the ontology. The hierarchy and various types of relations between ontology objects can be well plotted by a graph or a tree. An example of this group of methods is the tool OntoViz [20], which is a plug-in for Protégé. Disadvantages of using this technique are: a lack of interactions, problems with navigation, a lack of a searching tool and low effectiveness of utilization of the space on a display.

The *zoom-able techniques* visualize nodes from lower level inside their parent nodes. An example of the technique is CropCircles [19], which visualizes ontology in the form of hierarchy of concentric circles. This group of techniques is suitable for searching ontology with the purpose of finding a concrete object. These tools do not provide understandable picture of the ontology structure.

The *space-filling techniques* aim at the best utilization of a space. They divide the space available for some representational node into rectangles. Each rectangle belongs to one descendant of this node (for examples TreeMaps for two and three dimensional space [24]). Disadvantage of them is a lack of space remaining for internal nodes. They are inadequate for an ontology structure visualization.

The *focus + context or distortion access* is based on the combination of a focusing method and a context. Usually, one node is in the centre and other nodes are situated around. Because of hyperbolic transformation, the larger distance is between those nodes, which are situated near the centre. Typical representatives of

this access are 2D Hyperbolic Tree and 3D Hyperbolic Tree [21], constituting a tree structure in two or three dimensional space. The tree root is situated in the centre and its descendants are placed around. When an actual node is changed, the tree visualization is rearranged around a new centre. It is suitable for providing a global view on ontology. Disadvantages of such methods are incompleteness of information about some of the nodes and continual redrawing of the graph.

The *3D information landscapes* methods locate ontology into a map and in this way define the context of the given ontology and its relation to other ontological documents on this map. The collocation is provided on the basis of relations between ontology and map components. The map can represent one or more domains. The examples of systems belonging to this category are File System Navigator (FSN) and Harmony Information Landscapes (HIL) [9]. The nodes are represented by three dimensional objects located on the map. Attributes of ontology documents are coded by colour and size of the given objects.

We were inspired by a metaphor of this map. Information in an ontology is usually too extensive to be visualized globally in its whole complexity. So we were motivated to design a visualization method, which allows information filtration and focusing on key concepts of the ontology. Our aim was to enhance readability and fast orientation within the ontology to improve the user navigation in the ontology space.

3   Ontology Evaluation

Ontology evaluation is a process of the determination of a measure in which a given ontology meets some defined criteria [2]. This process is often specialized to be able to identify a domain, the ontology logically belongs to. This domain may be covered by a given ontology in different measure and with different level of granularity. Known ontology evaluation methods can be divided into the following approaches: an approach based on *comparison with a gold standard, evaluation of results of applications using a given ontology, comparison with data sources,* and *human evaluation*. Another dividing of the evaluation methods according to the evaluation level is following: *lexical-data level, hierarchically-taxonomical level, semantic relation level, contextual-application level, syntactic level and architecture and design level*. Table 1 illustrates relations between these six evaluation levels (the first column) and four previously mentioned approaches to ontology evaluation (the first row). In case, that a relation exists (for example between lexical data and gold standard), the related cell is marked with "X".

The l*exical-data level* evaluation techniques focus on the concepts, instances and facts in the ontology. In [15], the evaluation technique is to describe the measures of similarity of two strings by a number from interval [0,1]. Each string from the

first set is compared with each string from the second set. In principle, this technique represents a comparison of all the names of concepts from a given ontology with the concepts of a gold standard. The gold standard can be represented by a group of strings being considered as a good representation of concepts from the given domain. The gold standard can be another ontology, concepts generated from text documents or defined by experts in the given domain.

The *hierarchically-taxonomical level* evaluation does not focus on the analysis of the objects (as previous access), but focuses on analysis of the structure of relations between these objects.

Table 1

Survey of ontology evaluation techniques - source [2]

Level	Ontology evaluation approach			
	Gold standard	Application based	Data comparison	Human based
lexical-data	x	x	x	x
hierarchically-taxonomical	x	x	x	x
semantic relation	x	x	x	x
contextual-application		x		x
Syntactic	x			x
architecture and design				x

The *semantic relation level* evaluation includes all types of semantic relations. Very often, it contains precision and recall computing.

The *contextual-application level* evaluation techniques focus on a context creation and evaluation in the framework of a real application. Various ontology documents can have mutual relationships between their parts or concepts. The relationships enable to connect given ontologies into one model and to create a formal and consistent domain description. This is the way, a context can be created. The ontologies are not intended for direct interactions with users. They are in the form, which is intended for reading by machines and common people (not experts) would have problems to read them. They are primarily designed for using in applications as an auxiliary source of information. Therefore, the quality of a used ontology influences the results of these applications and similarly good results of the application entitle us to presume good ontology quality. An approach for calculating the distance between two ontology concepts is described in [23]. The results are compared with a gold standard provided by an expert.

The *syntactic level* of evaluation is focused on manually created ontologies. These ontologies are written in some particular programming language. They fulfil the specifications of the used language. This fact can be utilized within the testing of the ontologies.

The *architecture and design level* is processed manually, mainly in the case when the ontology has to fulfil some predefined criteria.

4   New Evaluation Method Using Visualization

The main objective of our work is to navigate users in a large ontology space and help them to select a suitable ontology for their needs, interests or systems. The mentioned objective belongs to the field of user personalization and personalized web recommender systems [26]. To achieve the objective we decided to use ontology evaluation methods. This approach can be successful only with the aid of an effective ontology visualization method which enables to create a smart and quick picture of the content of an examined ontology. Therefore, we decided to design a combination of ontology visualization and evaluation methods based on descriptive vectors, inspired by a metaphor of 3D information maps. We have chosen this model, because it enables not only convenient user navigation in a large ontology space, but also it enables to express relations and even proximity of particular ontologies. The measure of the closeness or even of the diffusion/interleaving of ontologies in the 3D information map intuitively expresses the measure of semantic similarity. This property distinguishes our approach from other approaches.

The existing visualization techniques are too complex and thus they are inadequate for quick ontology searching and evaluation. We have designed an approach enabling reusing an ontology in a specific application even if the ontology was developed for a different purpose. Our approach enables an effective search of information within ontologies. Main steps of our design of the ontology visualization process are following:

- generating of a vector description of a domain
- generating of a vector description of an ontology
- comparing the two descriptive vectors
- visualizing in a context
- navigation in an ontology space.

4.1   Vector Description of a Domain

The aim of the vector description of a domain is to summarize all available information about the domain and to insert them into the vector in the compressed form. Each domain can be represented by one so called descriptive vector. The vector can be compared with an ontology descriptive vector to evaluate the measure of coincidence. The concept "domain" can be defined as a field of knowledge represented by entities, their relations, attributes, their values and rules, which associate elements on the higher level of generality. Formally

consistent sources of information about some domain can be just ontologies. Therefore we decided to use domain-oriented ontologies as sources of information for acquisition of the descriptive vector of a given domain. There are the following preconditions in regard to ontologies:

- the natural language used to define ontologies is English
- the syntactical properties of English are exploited in searching on ontologies
- the label plays the role of the denotation for a concept as a node in a complex network and it is not used for edges that represent relationships.

Some concept within the ontology can be represented by its label – name pair as well as by other concepts within its environment (consisting of the closest concepts). They both define concept's semantics. For example the concept "soul" accompanied by an environment "music, blacks, rhythm" has different semantics as the concept "soul" accompanied by the environment "spirit, psyche, animus". The representation of the context of a term (including its environment) is a vector of words – labels of concepts. The existing visualization techniques visualize ontology content in the form of a complex and complicated graph. That is why we have come with a solution, which can compress data from different ontologies and consequently about domain, in the form of domain descriptive vector d_i (1):

$$d_i = [(c_{i1}, w_{i1}), ..., (c_{iM}, w_{iM})]. \qquad \textbf{(1)}$$

Symbols w_{ik} are weights of the concepts c_{ik} with relation to the domain d_i. The number of domains is N: $i \in [1,N]$. Each of these vectors represents "gold standard" of the given domain. Unlike a classic gold standard, which was created manually, the gold standard within our approach was created in an automatic way by analysing contents of the related ontologies.

Within the descriptive vector of a given domain creation, all relating ontologies with respect to the domain have to be searched. At the beginning of this process, user has to enter a key term characterizing the given domain. The key term (key concept) can be an object of Class type. It cannot be an object of Individual or Property types. The Semantic web browser finds all the ontologies, containing this key term. All concepts from the nearest environment of the key term in the given ontology are selected and saved with their status. The information will be used for weight calculation for the descriptive vector of the given domain. Figure 1 illustrates examples of the ontologies, which were selected, because they contain the key term "*academic employee*" (red colour).

The nearest environment of the key term "*academic employee*" in our ontology example can be found in the left part of Figure 1 (green colour). It contains all nodes (*owl:Thing, lecturer, PhD student*), which are related directly to the original key term. The given key concept was found also in the ontology example on the right side of Figure 1 in the form of the term "*academic*", with its nearest environment (*academy, professor in academy, researcher in academy*). Our

approach considers also a partial match between the key term given by the user ("*academic employee*") and the key term found in the ontology ("*academic*"). All selected terms are inserted into the descriptive vector together with their weights. The weight of a term represents its semantic closeness to the key term. The weight is calculated in the following way. At first, initializing weights for key term (exact match, *academic employ*) and for similar terms (partial match, *academic*) are calculated.

These terms (red colour in Figure 1) are marked as *original concepts*. Each of the considered ontologies contains one original concept. The weight initialization of the original concept is calculated according to the type of the given object:

- object Class: $w_0 = 10 + G$
- object Individual: $w_0 = G$
- object Property: $w_0 = 1$.

where the object "Class" represents a group of similar "Individuals" and object "Property" represents some relation, for example some relation between classes. Intuitively, the object Class has greater weight than for example the Individual, because it contains more individuals and therefore it represents a concept, which is generally more valuable for visualization.

The coefficient $G \in [1,10]$ represents a generality level of the concept, where $G = 1/10$ represents minimal/maximal generality of the concept.

A *superior concept* is the concept containing a link to the original concept (the original concept is a type/subclass of the superior concept). The weights of superior concepts from the nearest environment of the original concept (*owl:Thing* for original concept *academic employee* in Figure 1) are calculated according to (2):

$$w = \frac{w_0 + G}{l}.$$ (2)

The parameter l is the number of words in the label of the concept. For example, "owl:Thing" has $l=2$ and "MusicalExpression" has $l=1$ (Figure 3). The way of computation of this parameter ensures the preference of concepts with smaller number of words in their labels. These concepts have higher information value. On the other hand, the Class - concept that is labelled with multiple words has lower information value and also lower weight. The parameter "l" is not taken into account into original weight w_0, because w_0 and l are two parameters used in computing the final weight w. The original concept that matches only partially to the given key concept gets the different weight as the fully matching original concept.

An *inferior concept* is the concept containing a link from the original concept (the inferior concept is a type/subclass of the original concept). The weight of inferior concepts from the nearest environment of the original concept (*lecturer, PhD*

student for original concept *academic employee* in Figure1) is computed according to the formula (3):

$$w = \frac{w_0}{l}.$$

(**3**)

Both, superior and inferior concepts take into account the existing hierarchy of the ontology. The main difference between the superior and the inferior concepts is that the superior one contains "a link to" and inferior one contains "a link from" the original concept. They cannot have the same weights as the original concept, which is the core of the descriptive vector and which is more similar to the key word, even if their labels contain the same number of words.
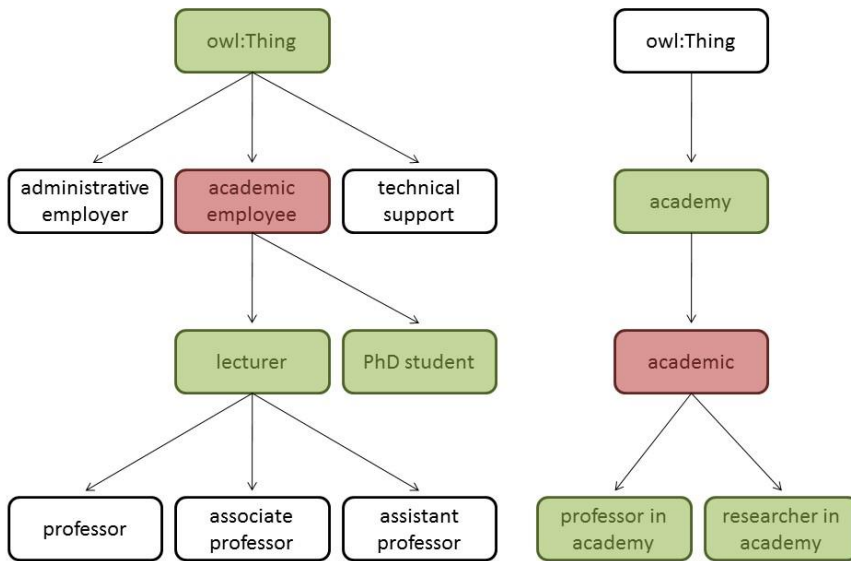
Figure 1

Tree-like graphs of two ontologies (type of using relation is "subClassOf")

For the key concept "a*cademic employee"* and *G=1* the following concepts from the ontology on the left in Figure 1 are collected into the domain descriptive vector:

[(academic employee, 11), (owl:Thing, 12), (lecturer, 11), (PhD student, 5.5), ...].

The numbers in this vector represent weights of the vector concepts. The change of the parameter *G* into value *G=10* leads to the following modification of the descriptive vector:

[(academic employee, 20), (owl:Thing, 30), (lecturer, 20), (PhD student, 10), ...].

In the case of higher occurrence of labels consisting of more words, the difference of vector modification is more significant. The descriptive vector is only a particular vector, which was created from the left ontology in Figure 1. The whole domain descriptive vector comes into being by aggregation of all particular vectors coming from all ontologies containing the given key concept. The aggregated domain descriptive vector represents summary of all particular vectors, which were derived from the related ontologies.

Various particular vectors can contain the same concept but with different weights. The concept is inserted into the aggregated vector only once with the weight, which is aggregated from all weights of the given concept coming from all particular descriptive vectors.

The next step is normalization and reduction of the domain vector. The normalization represents transformation of all weights into the interval [0,1] for the purpose of future comparison of the domain descriptive vector with an ontology vector. Consequently the concepts with lower weights than a given threshold T are eliminated from the vector (experimentally was stated the threshold T = 0.0005).

4.2   Vector Description of an Ontology

The vector description of an ontology obtains descriptive vectors of the key concepts of the ontology with their weights. The concepts on the most abstract levels of the ontology are not suitable for the role of the key concepts, because they are too general. Similarly, the concepts on the lowest levels of the ontology are too specific for common users. The most suitable and informative levels are those in the middle of the ontology taxonomy. This idea was used in a method developed in Knowledge Media Institute in Open University in Great Britain within the project NeOn [4]. The method is based on the search of *n* concepts, which describe the ontology in the best way – key concepts of the ontology. The method tries to maximize centrality of the concept (maximum number of appearances in all paths from the root of the ontology) and to minimize the number of words in the concept label. In addition, the method tries to maximize the density of the concept (the number of concept instances or its frequency) and the concept coverage (the number of other key concepts in the ontology, which belongs to the sub tree of the given concept). All key concepts with the highest information value represent the ontology summary. The descriptive vector of the key concept contains also concepts from its environment and it reflects only one ontology context. The relevant concepts for the inclusion into the descriptive vector of a key concept are all ancestors and all descendants of this key concept as illustrated in Figure 2.

The weights of concepts in the descriptive vector are calculated according to formulas (2) and (3) with the initializing weight value equivalent to "10" and "G=5". For example in Figure 2 the description of the key concept "*supervisor*" (in the form of the descriptive vector) is:

[(supervisor, 15), (agent, 20), (owl:Thing, 10), (professor, 15), (senior researcher, 7.5), (assistant professor, 7.5), (associate professor, 7.5)].

The descriptive vector of each key concept must be normalized into interval [0,1] for the purpose of enabling subsequent comparison with a domain descriptive vector. The terms with weights lower than the threshold value *T=0.0005* are eliminated from the key concept vectors. After carrying out the mentioned steps, the descriptive vector of the key concept "supervisor" will be the following:

[(agent, 1), (supervisor, 0.75), (professor, 0.75), (owl:Thing, 0.5), (senior researcher, 0.375), (assistant professor, 0.375), (associate professor, 0.375)].
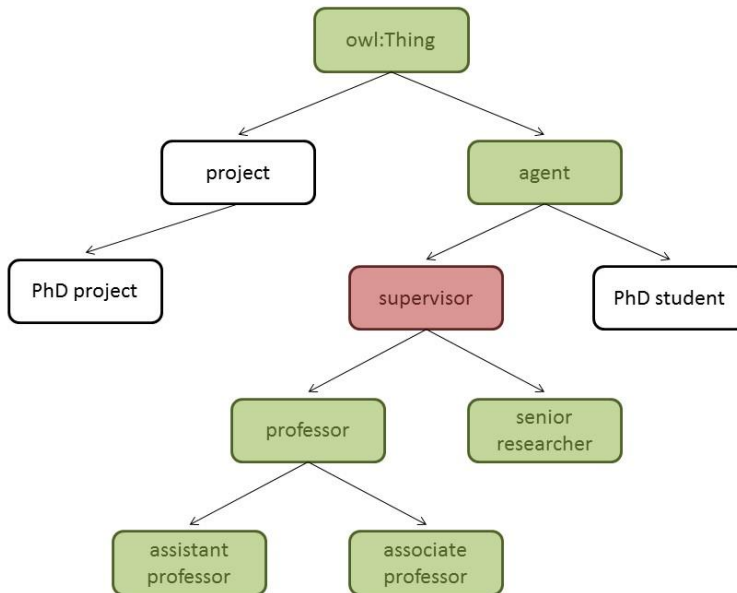

Figure 2

Key concept (red colour) of the ontology together with its relevant concepts (green colour). In this case, not only the nearest environment of the key concept is taken into account but also the rest of antecedents and descendants of the key concept, because the nearest environment would be represented by only three nodes

4.3   Comparison of Descriptive Vectors

Our approach uses the well known cosine metric of the similarity between the vectors of the ontology and the domain. According to [14] the cosine metric of the similarity is suitable for short texts. The metric expresses a cosine of the angle between the two vector representations in the coordinate system – the domain descriptive vector and the vector of the context of the ontology. The key concepts are located in the domain space and their coordinates are determined by their similarity measure within the given domain. Each of the similarities $S(x_i, x_j)$ is calculated, where x_i is vector of the *i-th* ontology (i $\in$ [1,M]) and x_j is vector of the *j-th* domain (j $\in$ [1,N]). The resulting similarity matrix is following:

$$S = \begin{bmatrix} S_{11} & S_{1M} \\ S_{N1} & S_{NM} \end{bmatrix}.$$

(4)

The similarity matrix can be used for various purposes. For example, with the aid of the similarity matrix, the best location of the ontology key concepts in the domain space can be assigned within the visualization method. Other application possibilities are an automatic ontology evaluation and an automatic ontology searching according to user requirements and criteria. One of such criteria can be represented by a set of domains, which must be covered by the given ontology in the significant measure. Another criterion can be searching for only one domain covered by the given set of ontologies in the best way. Next possible applications can solve the problems of ontology comparison, combination of ontologies into larger information systems or key words extraction from a domain. We do not focus on the WordNet, because we utilize a content of all available ontologies within the web space.

5   Implementation of the Designed Method

From the view of our implementation of the designed method, two tools for acquisition of the ontological data were considered: Swoogle [6] and Watson [3]. Swoogle is an older tool and it cannot distinguish different versions of an ontology. The tool also administrates ontologies only on the level of documents and it cannot provide functions for access to objects into the given ontology. On the other hand, Watson can distinguish various versions of the same ontology and can manage accesses to the stored ontologies and enables their reusing, which is a very important feature. On the basis of these facts, the tool Watson was selected.

The designed method - combination of ontology visualization and evaluation based on descriptive vectors - was implemented as a system called OntoSumViz

and subsequently tested. This system works in three steps: semantic content acquisition, concepts processing and cache filling.

The *semantic content acquisition* was executed with the aid of the Watson system using Java Client API. The module downloads the ontologies containing a given key word. It extracts sub-trees (which contain the key concept) from the searched ontologies and its nearest environment together with information about their types and relationships. The sub-trees are assigned to the original concept found in the given ontology.

Within the module of *concepts processing*, the weights are initialized. In the next step, the weighting scheme is applied for modification of concepts weights, aggregation and normalization. The resulting concept descriptive vector is an input for the comparator. It compares the descriptive vectors of given domains and the descriptive vector of a key concept in an ontology and subsequently allocate the domain for the given ontology.

In the last step a *cache is filled* in order to save descriptive vectors for next reuse. The cache module checks whether the descriptive vector for the key concept given by a user occurs in the buffer. Only in the case when it was not found, the implementation OntoSumViz starts computing of a new descriptive vector. To inspect the number of the ontologies containing the given key concept, it is necessary to call the special service, which takes 11 seconds, which is an average value of the time response of the special service. It illustrates the system as extra time consuming. For the vector with 200 terms, the service has to be called 200 times. It represents a considerable delay. Therefore we decided for another solution. It is using the vectors from the cache in the role of the corpus of the ontologies.

Our implementation of the designed method is realized as a module of OntoSumViz. The novelty of our approach is the design of descriptive vectors computing. This approach offers quick overview of the given ontologies content without long searching and exploration.

5.1   Visualization of the Context within the OntoSumViz

The approach, which was described within previous sections, can be used in many ways. We use it for creating the context of the ontology. It was mentioned, that the concepts with the significant information value are situated in the middle level of the ontology. Our implementation offers these concepts to user automatically within the ontology visualization. The implementation characterizes the meaning of the concepts with the aid of their environment and in this way it makes easier the decision making process of the user about suitable ontology. A principle of gradual uncovering of the ontology content is consistent with user mental model creation.

The whole number of middle level concepts in the visualized ontology is divided into three levels of significance. In each level, the same number of concepts occurs. In the case, when twelve concepts are added on the sheet, these twelve concepts are divided into three significant levels and each level contains four concepts. The process of the visualization of these concepts is the following: at the first moment, only the four most important concepts for the decision about the ontology suitability and the measure of interest of the given ontology for a user are visualized. Next, other four less important concepts are displayed and, at the end, the four least important concepts are visualized.

5.2   Ontology Location in the Context

Within our implementation, the ontology is represented by a set of key concepts. The user can change the level of details by closing in respectively secluding (zooming out) the used view. To specify the meanings of the key concepts, the metaphor of a geographical map is used. The map is represented by the space, which is divided into sectors (9 sectors in Figure 3). The sectors represent different domains – different possible user's interests. The ontology can (intuitively) exceed the borders of one domain. The key concept is defined by its environment on the map. Visualization of such a map and visualization of the selected ontology at the same time are illustrated in Figure 3. The implementation of the OntoSumViz is a component of the tool NeOn Toolkit [8], [16].

The domains (their names use capital letters) are represented by characterizing concepts (using small letters) in the related sectors. The terms originate from the descriptive vectors of the given domains. The descriptive vector of the domain cannot be displayed completely because of its cardinality. Due to this reason, only six the most important terms are displayed. As an example, the key concepts of the ontology "musicontology.rdf" (including their environment) are located onto map (yellow colour) on the basis of the similarity calculation between the descriptive vector of the domain (nine descriptive vectors are used, one for each domain in the example in Figure 3) and the descriptive vector representing the ontology key concepts. The ontology key concepts are situated in those domains, whose vectors are the most similar to the given key concepts vector. The ontology key concepts are situated in the positions, which are nearest to the most similar concepts of particular domains.
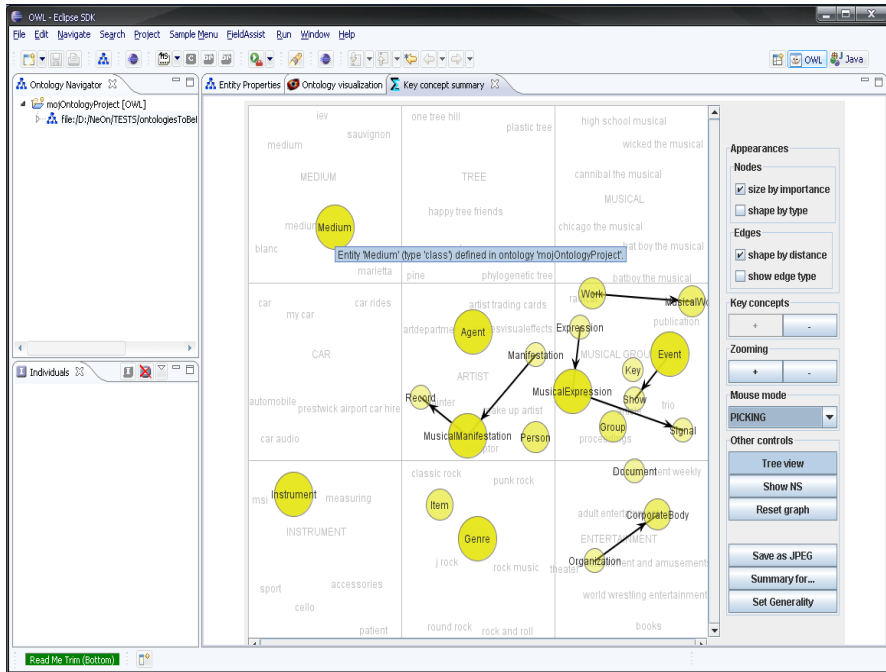
Figure 3

The screenshot of the NeOn Toolkit [16] with a map and with visualization of an example ontology using OntoSumViz (right window - control panel of the OntoSumViz)

The ontology key concepts can be distributed to more than one domain. Figure 3 illustrates distribution of the ontology key words into 6 domains, because relations to the other three domains are marginal insignificant. Thus, it can be said, that the ontology belongs mainly to two domains: ARTIST and MUSICAL GROUP. If there is some direct relation between two concepts, then this relation is displayed by an arrow (see Figure 3). The presented implementation can be used also for displaying more ontologies on the same map.

5.3   Navigation in the Ontology

Navigation in the ontology is performed using the control panel of the tool OntoSumViz illustrated in the right window in Figure 3. The panel contains buttons, which are grouped into the following blocks: Appearances, Key concepts, Zooming, Mouse node and other controls.

Within the block "Appearances", presentation of nodes and edges of the ontology graph can be set up. The block "Nodes" contains two possibilities: "size by importance" and "shape by type". The size by importance selection enables to set a size of displayed key concepts according to their importance. The key concepts,

which are displayed on the first significance level of approaching (see Section 5.1) have higher importance and therefore are of bigger size than the key concepts from other significance levels. The shape by type selection distinguishes key concepts according to types. A key concept of the type class is represented by a circle and a key concept of the type instance is represented by a square with rounded edges. The block "Edges" contains two possibilities of edge form setting: "shape by distance" (a thick link represents the relation between concepts (subClassOf) and a thin link represents the relation between class and instance (instanceOf)) and "show edge type" (each link is signed by its name and type).

The blocks "Key concepts" and "Zooming" enable application of the metaphor of a geographical map. User can enlarge some part of the map and see this part in more details. The block "Zooming" provides traditional closing (buttons "+" and "-") and the block „Key concepts" provides contextual zooming – contextual navigation, when more detailed view of particular concepts is provided. The higher/lower level of significance can be achieved by buttons "+"/"-".

The block "Mouse node" enables manipulation with the whole graph ("transforming" – implicit setting) or moving one node of the ontology graph in the case, when two concepts (nodes) overlap ("picking" selection).

The last block "Other controls" enables access to the following menu possibilities:

1.   "Tree view" button switches between the ontology displaying on the map and its displaying in the form of a graph.

2.   "Show NS" button shows the whole name of the key concept.

3.   "Reset Graph" button reinitializes ontology and locates it into the map in the case when some changes were performed, for example changing of the parameter generality.

4.   "Save as JPEG" button enables saving of the map or the graph in the form of a picture.

5.   "Summary for" button sets actual user.

6.   "Set Generality" button" enables setting of parameter G from interval [1,10] (as an implicit setting is used the value 5).

6   Experimental Analysis

A set of experiments with the OntoSumViz implementation was performed with the aim to verify the designed methods. The tests were focused on the following issues:

- possibility to use the vector description of a domain as a golden standard of the given domain,
- precision of the designed methods and effectiveness of the implementation OntoSumViz within the user navigation in an ontology space,
- comparison of the decisions provided by the implementation OntoSumViz against decisions of experts.

6.1   Vector Description of a Domain as a Golden Standard

The golden standard of some domain is an etalon of the domain, which can be created by some expert in the given domain. We wanted to know, if our implementation OntoSumViz can be applied for building this golden standard and how many ontologies are necessary to be used for this golden standard building. In our case the golden standard would have the form of a set of concepts - items of the descriptive vector of the domain.

Some experiments concerning the golden standard were performed. The experiments were carried out to find an optimal number of ontologies needed for computing the descriptive vector on the satisfied level of precision. We tried to verify also suitability of using the generated descriptive vector of a domain as a golden standard. We have performed a series of six experiments with various numbers of ontologies used for developing a descriptive vector (MaxOnt = 10, 100, 200, 300, 400, 500 – MaxOnt is the maximal number of the used ontologies) which can be seen in rows of Table 2. The experiments showed the degree of matching (values in the cells of the table) between one of the three domains "Academic Employee", "Project", "Object" and a set of seven domains in columns of the table ("Instrument", "PhD Project", "Student", "Education", "Music", "Supervisor" and "Entertainment"). The values in this table represent particularly the cosine similarity metric between two vectors of two domains. The darker shade of colour represents higher degree of matching between the two given domains. The highest degree of matching within these experiment can be seen between the domain "Project" and "PhD Project" (e.g., in the experiment with MaxOnt=400, the similarity matching value equals to 0.8204).

It can be seen, that the increasing number of the ontologies, used for the domain descriptive vector calculating, causes that the values of similarity are more precise and less diffused and the position of some domain from the above given ontology triplet in some column is reinforced. At the same time, the position in the other columns is weakened. The values of parameter MaxOnt, which are higher than 300, do not cause any significant change of the results. Thus, just the value MaxOnt=300 seems to be an adequate compromise between time complexity of the calculation and precision.

Table 2

Degree of matching between one of domain "Academic Employee", "Project" and "Object" and the seven selected reference domains ("Instrument", "PhD Project", "Student", "Education", "Music", "Supervisor" and "Entertainment")

Academic Employee

	Phd Project	Student	Instrument	Education	Music	Supervisor	Entertainment
10	0,2550	0,0997	0,0000	0,0210	0,0000	0,0577	0,0007
100	0,4308	0,0632	0,0392	0,0159	0,0019	0,2623	0,0000
200	0,2314	0,0578	0,0595	0,0338	0,0102	0,1494	0,0614
300	0,2374	0,0571	0,0595	0,0000	0,0093	0,2128	0,0526
400	0,2350	0,0512	0,0377	0,0380	0,0064	0,2113	0,0587
500	0,3200	0,0389	0,0270	0,0836	0,0069	0,2276	0,0509

Project

	Phd Project	Student	Instrument	Education	Music	Supervisor	Entertainment
10	0,8602	0,0433	0,0127	0,0000	0,0131	0,0193	0,0180
100	0,8371	0,0482	0,0038	0,0138	0,0043	0,0282	0,0088
200	0,8838	0,0251	0,0049	0,0080	0,0045	0,0091	0,0100
300	0,8475	0,0129	0,0026	0,0100	0,0021	0,0046	0,0058
400	0,8204	0,0142	0,0033	0,0030	0,0020	0,0034	0,0043
500	0,7583	0,0099	0,0021	0,0025	0,0008	0,0035	0,0036

Object

	Phd Project	Student	Instrument	Education	Music	Supervisor	Entertainment
10	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
100	0,0033	0,0007	0,1003	0,0456	0,0233	0,0003	0,0330
200	0,0038	0,0031	0,0900	0,0823	0,0344	0,0014	0,0920
300	0,0050	0,0132	0,1366	0,0999	0,0477	0,0024	0,1084
400	0,0062	0,0261	0,1556	0,1182	0,0543	0,0047	0,1238
500	0,0062	0,0360	0,1625	0,1135	0,0538	0,0053	0,1264

6.2   OntoSumViz Implementation Testing

The main goal of the implementation OntoSumViz is to navigate users in the ontology space and to help them to select a suitable ontology for a given application or a given problem. Therefore we performed tests to compare the precision of the user navigation by experts and by the implementation OntoSumViz. Three different experts have determined the ontology key concepts (the first column of the Table 3 – "Genre", "Expression", ...) belonging to the defined domains (the first row of the Table 3 – "Artist", "Entertainment", ...). At

first, semantic matching between the key concepts and domains was determined by all experts as a number from the interval [0,5]. Next, the measure of agreement of all experts was quantified by the standard deviation (Bessel modification was used). The results of the test are illustrated in Table 3.

Table 3

Standard deviation of experts' agreements in the determination of belonging of the given key concepts to the selected domains

	Artist	Entertainment	Musical Group	Rock	Medium	Musical	Instrument	Car	Tree
Genre	2,645751311	2,645751311	2,645751311	2,886751346	1,154700538	1,154700538	2,081665999	0	0
Expression	2,886751346	2,645751311	2,081665999	2,081665999	2,081665999	1,154700538	2,081665999	0	0
Person	0	2,886751346	0,577350269	2,886751346	2,886751346	1,732050808	0	1,732050808	0
Signal	0,577350269	1,527525232	1,732050808	1,527525232	2,645751311	2,309401077	2,309401077	0	0
Corporate Body	1,154700538	1,527525232	1,527525232	0,577350269	2	2,081665999	0,577350269	2,886751346	0
Release Type	2,081665999	1,527525232	2,081665999	2,081665999	1,732050808	2,081665999	0,577350269	0	0
Musical Manifestation	2,081665999	1,732050808	2,081665999	2,309401077	1,527525232	0	1,732050808	0	0
Manifestation	1,154700538	0,577350269	1,732050808	2,309401077	1,732050808	2,309401077	0,577350269	0	0
Musical Expression	1,527525232	1,527525232	1,527525232	2,081665999	1,527525232	0	1,732050808	0	0
Item	0,577350269	0,577350269	0,577350269	0,577350269	2,309401077	2,886751346	1,732050808	0,577350269	0
Musical Work	2,081665999	2,081665999	1,527525232	2,309401077	1,154700538	0	0,577350269	0	0
Work	2,886751346	0,577350269	2,309401077	0,577350269	0,577350269	0,577350269	0,577350269	0,577350269	0
Group	2,645751311	1,732050808	0	1,154700538	1,732050808	0	1,154700538	0	0
Document	2,309401077	1,154700538	0	0	1,732050808	2,516611478	0,577350269	0	0
Medium	2,309401077	0,577350269	1,154700538	0,577350269	0	2,645751311	0,577350269	0	0
Record	1,154700538	2,645751311	0,577350269	0,577350269	0,577350269	0	1,527525232	0	0
Agent	0	1,154700538	2,309401077	0	0	1,732050808	1,154700538	0	0
Show	1	0	1,527525232	1,154700538	0,577350269	0,577350269	1	0	0
Instrument	0,577350269	1,527525232	1	1,154700538	0,577350269	0	0	0	0
Instant	0	0	0	0	0	0,577350269	0	1,732050808	0
TmeLineMap	0	0	0	0	0	0	0	0	0

The standard deviation shows differences among the decisions of the particular experts. Value 0 represents absolute agreement of all experts. The experiment acquired also pairs of the key words of the domain with a clear assignment for example "Instrument" – "Instrument". Such pairs can show clear agreement between experts (Table 3). There are some domains without any relation to the ontology key concepts for example "Tree" and partially "Car". They can show that concepts are not assigned in a random way. Another tested example is the case, when one key concept can be assigned to several domains and not belonging to any one from the given domains clearly (e.g., "Genre"). The experiment proved that the agreement among experts in this case is not very high. Nevertheless, values in the Table 3 are better than we expected.

6.3   Comparison of the Implementation OntoSumViz with Experts

For the purpose of another experiment, an arithmetic average of experts' responses was computed. The obtained average values were transformed into the interval [0,1]. This step cannot be omitted – it is necessary for the results to be comparable with the results obtained from OntoSumViz. Next, decisions of the implementation OntoSumViz were collected. They are also from the interval [0,1]. This interval represents cosine similarity metric, which has two extreme values: 0 (represents absolute dissimilarity) and 1 (represents absolute similarity). The absolute similarity is only theoretical, because the domain descriptive vectors have usually significantly higher cardinality than a descriptive vector of ontology key concepts. Finally, the differences of experts' average values and the OntoSumViz's values were calculated (Table 4). The results of this comparison are numbers from the interval [-1,1].

Table 4

Comparison between responses of the exports and the implementation OntoSumViz

	Musical	Artist	Musical Group	Entertainment	Rock	Instrument	Medium	Car	Tree	Average
Record	1	0,717933333	0,666666667	0,6	0,733333333	0,533333333	0,728533333	0	0	0,553311111
Group	1	0,5975	0,7644	0,6	0,866666667	0,458666667	0,4	0	-0,0004	0,520759259
Show	0,862966667	0,8	0,709533333	0,9906	0,733333333	0,3853	0,065966667	-0,0016	0	0,505122222
Musical Expression	0,9917	0,659566667	0,654866667	0,666666667	0,533333333	0,6	0,266666667	0	0	0,485866667
Instrument	0,9553	0,666666667	0,6659	0,465466667	0,666666667	0,6639	0,066666667	0	0	0,461174074
Genre	0,864366667	0,5988	0,583	0,6	0,472066667	0,533333333	0,133333333	0	-0,0006	0,420477778
Musical Manifestation	1	0,520533333	0,533333333	0,6	0,466666667	0,4	0,262666667	0	0	0,420355556
Expression	0,859766667	0,659366667	0,458866667	0,4	0,466666667	0,333333333	0,466666667	0	0	0,404962963
Work	0,858566667	0,665166667	0,504533333	0,533333333	0,133333333	0,732633333	0,133133333	0,133333333	0	0,410448148
Musical Work	0,99	0,531933333	0,433366667	0,666666667	0,466666667	0,265966667	0,133233333	0	0	0,387537037
Manifestation	0,733333333	0,520033333	0,6	0,466666667	0,466666667	0,333333333	0,1959	0	0	0,368437037
Person	0,1949	0,9177	0,866666667	0,327933333	0,333333333	0	0,333333333	0,2	0	0,352651852
Agent	0,5988	0,9617	0,509633333	0,266666667	0	0,261966667	-0,0034	0	0	0,288374074
Signal	0,531133333	0,066666667	0,194	0,333333333	0,266666667	0,533333333	0,6	0	0	0,28057037
Medium	0,6	0,533333333	0,128733333	0,066666667	0,066666667	0,066666667	0,7313	0	0	0,243707407
Release Type	0,466666667	0,333333333	0,333333333	0,266666667	0,466666667	0,066666667	0,2	0	0	0,237037037
Corporate Body	0,465666667	0,133333333	0,306533333	0,281933333	0,066666667	0,066666667	0,3957	0,331433333	0	0,227548148
Item	0,331133333	0,066666667	0,064466667	0,066666667	0,066666667	0,1976	0,264966667	0,064966667	-0,0024	0,124525926
Document	0,527533333	0,239566667	-0,0063	0,096433333	-0,006	0,062766667	0,1953	-0,0124	-0,0026	0,121588889
Instant	0,066666667	0	0	0	0	0	0	0,2	0	0,02962963
TmeLineMap	0	0	0	0	0	0	0	0	0	0
Average	0,932518519	0,653948148	0,615677778	0,60622963	0,563562963	0,515611111	0,280403704	0,014637037	-0,000111111	

Once again, darker shade of colour represents higher disagreement between the experts and the OntoSumViz. The shadowing illustrates ordering of the key concepts and the domains from more questionable to more clear. The last column and the last row contain "Average" values, representing classification errors.

Since there are only a few negative values, it is clear that experts's decisions are systematically higher than the decisions of OntoSumViz (its highest similarity value is 0.3361 only, while the highest similarity assigned by the experts was 1.0).

The most questionable domain is the domain "Musical". The ambiguousness of the domain causes discordance among experts. This fact has an influence on different classifications by experts and by the implementation OntoSumViz. The OntoSumViz takes into account all possible contexts of the word or phrase. On the other side, experts take into account only one context of the word, usually more probable according to their experiences. Very often the OntoSumViz system prefers a domain on the higher level of generality. The vectors of more general domains have usually higher cardinality and so higher chance to match with some key concept descriptive vector.

From the point of classification, 8 concepts from 21 were classified in the same manner (the experts and OntoSumViz agreed in their classification), i.e. the overall classification error was 0.619. The error is influences by the selected domains – the most problematic domain is "Musical". After removing this domain from the test, the number of correctly classified concepts increases to 12.

Conclusions

This work provides a new insight into the ontology evaluation field according to its suitability for solving a given problem. For example, such problem can be a decision in the form of selection of an ontology for a system Magpie [7]. The system needs to load a suitable ontology for a given domain. The semantics of the Magpie (explanation of concepts for user) is based on availability of such ontology. In contrast to the hierarchical tree representation, our approach visualises and interprets concepts with the help of the context, which is represented by their environment (the neighbourhood concepts). Thus, semantics of the concepts is essential within the process of the ontology visualization. Our approach implements the principle of conceptual basis in the form of a map, which helps users to discover those domains, which the given ontology covers. In case of the necessity to visualize more than one ontology, our approach helps users to see the main differences among considered ontologies.

The main contribution of this work is the design and implementation of the method of the vector descriptions of domains, which are generated by the information contained in the related ontologies. This approach offers quick overview of the given ontologies content without long exploration. We suppose, that ontologies are more valuable for this purpose than web pages or text documents, because ontologies contain dictionary of uniformly defined concepts, defined also by their properties and relations. All the facts were taken into account during the descriptive vectors design. Another very important contribution in the visualization field is placing the represented ontology on the map and creating the environment for the ontology. The novelty of the implementation OntoSumViz is

also in the combination of the ontology summarization with the conventional tree structure visualization.

We can see some possibilities for further extensions of the designed and implemented approach, for example looking through more ontologies at the same time. User could locate two different ontologies on the map and denote their mutual complementation or combination. It could be useful in the case, when the problem could not be satisfactory solved with the aid of a single ontology. In solving practical problems it is rather rare to find a single ontology, which is able to cover the whole problems and needs. For this reason it is very suitable to aggregate concepts from more sources.

Another extension could be enriching the visualization by the functionalities of the implementation to make it more helpful for those users, who require a more complex ontology view. The combination of visualization approaches could be suitable for the possibility to switch between different visualization views, for example between the contextual and the semantic view. It could be also interesting to investigate new application domains of the descriptive vectors, which have potential overlapping the field of the ontology visualization. One possible domain is the field of recognizing personality aberration from a written text [18], where the descriptive vector for each aberration will be created from the texts written by persons suffered from this aberration. Consequently it will be compared with the descriptive vector of some new patient.

Acknowledgement

References

[1]   Alatrish, E. S., Tošič, D., Milenkovič, N.: Building Ontologies for Different Natural Languages. In: *Computer Science and Information Systems* 11(2), 2014, 623–644.

[2]   Brank, D. M. J., Grobelnik, M.: A Survey of Ontology Evaluation Techniques. In *Proc. of the 8th Int. multi-conference Information Society*, 2005.

[3]   d'Aquin, M., Baldassarre, C., Gridinoc, L., Angeletou, S., Sabou, M., Motta, E.: Watson: A Gateway for Next Generation Semantic Web Applications. Poster session of the *International Semantic Web Conference*, 2007.

[4]   d'Aquin M., Motta E., Peroni S.: Identifying Key Concepts in an Ontology, through the Integration of Cognitive Principles with Statistical and Topological Measures, Knowledge Media Institute, 2006.

[5]   Daramola, O., Sindre, G., Moser, T.: A Tool-based Semantic Framework for Security Requirements Specification. *Journal of Universal Computer Science*, Vol. 19, No. 13, 2013.

[6]   Ding, L., et al.: Swoogle: A Search and Metadata Engine for the Semantic Web, Proceedings of the Thirteenth *ACM Conference on Information and Knowledge Management*, 2004.

[7]   Domingue, J. B., Dzbor, M., Motta, E.: Collaborative Semantic Web Browsing with Magpie, In Proc. of the 1st *European Semantic Web Symposium* (ESWS), Greece, May 2004.

[8]   Dzbor, M., Motta, E., Builarabda, C., Gomez-Perez, J. M., Goerlitz, O., Lewen, H.: Analysis of User Needs, Behaviors & Requirements with Interfaces and Navigation of Ontologies. Deliverable report D4.1.1, NeOn Project Consortium, 2006.

[9]   Eyl, M.: The Harmony Information Landscape: Interactive, Three Dimensional Navigation trough an Information Space. Graz University of Technology, Austria, 1995.

[10]   Flotyński, J., Walczak, K.: Semantic Representation of Multi-platform 3D Content. In: *Computer Science and Information Systems* 11(4), 2014, 1555–1580.

[11]   Garcia-Moreno, C., Hernandez-Gonzalez, M. A., Minarro-Gimenez, J. A., Valencia-García, R., Almela, A.: A Semantic-based Platform for Research and Development Projects Management in the ICT Domain. *Journal of Universal Computer Science*, Vol. 19, No. 13, 2013.

[12]   Gruber, T. R.,: A Translation Approach to Portable Ontology Specifications. *Knowledge Acquisition*, 5(2), 1993, 199-220.

[13]   Katifori, A., Halatsis, C., Lepouras, G., Vassilakis, C., and Giannopoulou, E.: Ontology Visualization Methods - A survey. *ACM Comput. Surv*. 39(4), 2007, 0-4.

[14]   Lee, M., Pincombe, B, Welsh, M.: An Empirical Evaluation of Models of Text Document Similarity. *Proceedings of the 27th Annual Conference of the Cognitive Science Society*. Mahwah, NJ: Erlbaum, 2005, 1254-1259.

[15]   Maedche, A., Staab, S.,: Measuring Similarity Between Ontologies. CIKM, LNAI vol. 2473, 2002.

[16]   NeOn Project. [Online]. Available: http://www.neon-project.org/nw/Welcome_to_the_NeOn_Project (current August 2015)

[17]   Nguyen, V. C., Qafmolla, X., Richta, K..: Domain Specific Language Approach on Model-driven Development of Web Services. *Acta Polytechnica Hungarica* Vol. 11, No. 8, 2014, 121-138, ISSN 1785-8860.

[18]   Ondrejka, A., Šaloun, P., Ceplakova, R.: Identification of a Personality Aberration from a Written Text. *Proc. of the 10th Workshop on Intelligent and Knowledge-oriented Technologies*, 2015.

[19]   Parsia, B., Wang, T., Golbeck, J.: Visualizing Web Ontologies with Cropcircles. *End User Semantic Web Interaction WS @ ISWC2005*, 2005.

[20]   Sintek, M.: Ontoviz tab: Visualizing Protégé Ontologies. [Online]. Available: http://protege.stanford.edu /plugins/ontoviz/ontoviz.html (current August 2015)

[21]   Souza, K., X. S., Dos Santos, A. D., Evahgeista, S. R. M.: Visualization of Ontologies through Hypertrees. In Proceedings of the *Latin American Conference on Human-Computer Interaction*, Rio de Janeiro, Brazil, 2003, 251–255.

[22]   Sram, N., Takács, M.: An Ontology Model-based Minnesota Code. *Acta Polytechnica Hungarica* Vol. 12, No. 4, 2015, 97-112, ISSN 1785-8860.

[23]   Superkar, K.: A Peer-review Approach for Ontology Evaluation. *Proc. 8th Intl. Protégé Conference*, Madrid, Spain. (2005)

[24]   Van Ham, F., Vanwijk, J. J.,: Beamtrees: Compact Visualization of Large Hierarchies. In *Proceedings of the IEEE Conference on Information Visualization*. IEEE CS Press, 2002, 93–100. (2002)

[25]   Xu, B., deFréin, R., Robson, E., Ó Foghlú, M.: Distributed Formal Concept Analysis Algorithms Based on an Iterative MapReduce Framework. [Online]. Available: http://link.springer.com/chapter/10.1007%2F978-3-642-29892-9_26#page-2 (current March 2016).

[26]   Zhu, T., Hu, B., Yan, J., Li, X.: Semi-supervised Learning for Personalized Web Recommender System. In: *Computing and Informatics*, 29(4), 2010: 617–627. (2010).