Webology, *Volume 15, Number 2, December, 2018*

Method for Ontology Content and Structure Optimization, Provided by a Weighted Conceptual Graph

Vasyl Lytvyn

Professor and Head of Information Systems and Network Department, Lviv Polytechnic National University, Bandery St., 12, Lviv, Ukraine. ORCID. E-mail: vasyl.v.lytvyn@lpnu.ua

Victoria Vysotska

Associate Professor and Deputy Head of Information Systems and Network Department, Lviv Polytechnic National University, Bandery Street, 12, Lviv, Ukraine. ORCID. E-mail: victoria.a.vysotska@lpnu.ua

Dmytro Dosyn

Associate Professor of Information Systems and Network Department, Lviv Polytechnic National University, Bandery Street, 12, Lviv, Ukraine. E-mail: dmytro.dosyn@gmail.com

Yevhen Burov

Professor of Information Systems and Network Department, Lviv Polytechnic National University, Bandery Street, 12, Lviv, Ukraine. ORCID. E-mail: yevhen.v.burov@lpnu.ua

Received June 12, 2018; Accepted December 25, 2018

Abstract

The complex dynamic structure of the ontology and its content requires implementing of optimization procedures to better ontology processing performance and resolve conflicts between data. This paper addresses the issues of ontology optimization with the purpose of adapting its content to the needs of users by excluding those items that are rarely used or not used at all, or which do not belong or not related to a particular subject area. The approach is based on automated weighting of concepts and relations during ontology learning. The ontology expanded in this way, is sequentially optimized according to criteria of integrity, absence of ambiguity, volume, response time, completeness and thematic balance. The optimization method of minimal spanning tree search was applied. This optimization task can be further reduced to the backpack problem for which the effective solving algorithms are known. The use of developed optimization techniques provides controlled automated ontology learning procedure that significantly expands the usability of ontology-based systems and reduces the time expenditures on their implementation.

Keywords

Ontology learning; Adaptive ontology; Conceptual graph; Knowledge base; Intelligent agent

---

Introduction

The continuing interest and growth of research efforts in the area of ontological modelling is fueled by untapped potential and numerous unresolved problems pertaining to systems based on knowledge (Basyuk, 2015; Chen et al., 2016; Su et al., 2017; Korobchinsky et al., 2017; Korzh et al., 2014; Fedushko, 2014; Vysotska, 2016; Lytvyn et al., 2016). However, recently the number of works researching the practical applications of ontologies to problem solving in different domains, is also on rise. This is an unmistakable sign of growing maturity of this area of research.

The application of ontology-based methods to problems and tasks of various domains requires the evaluation and subsequent ensuring of those methods efficacy and effectiveness. For this a number of measures should be developed along with the methods and algorithms for system optimization based on those measures as criteria.

The measures and optimization methods can help to solve the problem of ontologies comparison, or provide an optimal solution when applying ontologies to problems in some domain, or help us to build the effective procedure of ontology construction.

The previous research

One of the popular optimization problem is the evaluating a mutual alignment of several ontologies and changing them in order to increase their alignment level (Euzenat & Shvaiko, 2007; Maedche & Staab, 2002; Xue et al., 2015; Martinez-Gil et al., 2008; Hassan et al.,2006; Mohammad et al., 2007; Lytvyn et al., 2017a; Lytvyn et al., 2018; Vysotska et al., 2015; Vysotska et al., 2018). Several measures had been proposed for comparing the ontologies similarity. They generally fall in a few categories: syntactic, linguistic, using data analysis, graph-mapping and taxonomy-based. For example, Levenstein's distance (Euzenat & Shvaiko, 2007) is based on operations with string transformations. It calculates the number of operations such as inserting, modifying, deleting characters done during the string transformation. Jaro's distance (Maedche & Staab, 2002; Rashkevych et al., 2017; Tkachenko et al., 2018) calculates a number of common characters in strings and takes into consideration positions where they appears. Linguistic measures evaluate the level of similarity using information about synonyms, hypernyms, and antonyms from the dictionary. Taxonomy-based measures use only specialization relation from taxonomy to calculate the level of similarity (Khomytska & Teslyuk, 2016; 2017). Data based methods study instances of concepts in ontology (Zhezhnych & Markiv, 2018; Lytvyn et al., 2017b; Lytvyn et al., 2017c). If some concepts have the same instances, then they are similar. Graph-mapping compares ontologies represented as graphs using methods from graph theory.

For measuring alignment levels the measures of recall and precision are used (Xue et al., 2015; Rusyn et al., 2016; Abolhassani et al., 2006; Ahmad & Colomb, 2007; Gozhyj et al., 2018). Recall describes the fraction of correct alignments found in ontology compared to all correct alignments. However it does not state the number of false alignments. Precision measures a fraction of correct alignments in all number of alignments. Several methods were developed in order to increase alignment of ontologies. Typically those methods are based on linguistic or taxonomic measures. In (Xue et al., 2015; Martinez-Gil, 2008; Martin et al., 2009; Precup et al., 2013; Rami & Vlach, 2002; Ramirez-Ortegon et al., 2013; Solos, 2016) a genetic algorithm is proposed for solving this problem.

The second class of optimization problems, related to effectiveness of ontology usage for some specific domain problems, is illustrated by using ontologies as an intermediate layer while forming queries to databases (Calvaneze, 2013; Gottlob et al., 2011; Li & Heflin, 2010). In such system, queries are formulated using concepts and relations from ontology (Gottlob et al., 2011; Davydov et al. 2017). After that, they are rewritten as SQL queries. The effectiveness of result is evaluated using well known measures such as query complexity. The query optimization methods reduce complexity by using dependencies between concepts in ontology (Li & Heflin, 2010), such as semantic index (Calvaneze, 2013).

The task of using ontologies for query expansion was studied in (Alipanah et al., 2011). The goal was to find concepts in ontology, which are semantically similar to concepts mentioned in query. In this way search should provide a larger selection of results. The authors in (Alipanah et al., 2011; Maksymiv et al., 2017) are defining the most k-top relevant terms using semantic metrics such as density measure, betweenness measure, and semantic similarity measure. In order to find an optimal solution a Map Reduce algorithm was used.

Ontologies are also used to optimize processes and task execution in specific domains. Typically, first of all, the processes and operations are studied and optimized for the domain. Next, obtained knowledge is formalized in form of ontology, which is later used for automation of processes and decision making. For example, the ontology-based system for data mining optimization is presented in (Keet et al., 2013; 2015, Peleshko et al., 2016). An exhaustive research on data mining decision-making patterns allows dynamically select optimization methods and criteria while processing data for authors of DMOP ontology.

The ontology optimality problem

The analysis of different methods and measures in ontology based optimization shows that often the same properties of ontologies are reused for optimization of different tasks and across domains. This warrants a further study of inner structure of ontologies in order to use it for optimization of tasks execution and problem solving in various domains. This paper addresses the issues of optimization for ontology management tasks, such as ontology creation, update, validation, storage and retrieval.

The usage of automated ontology development systems often results in the formation of defects in its structure and content, incompatibility of the ontology's content with the information needs of the user. The complex structure of the relationships between concepts represented in the ontology and its dynamic content, requires implementing of certain optimization procedures to minimize the response time to requests; satisfying constraints on storage volume reserved for the ontology; resolving logical conflicts between data that are extracted from a variety of sources, as well as satisfying other requirements and criteria that are to be determined. Optimization of ontology is also performed with the purpose of adapting its content to the information needs of users by excluding those items that are rarely used or not used at all, or which do not belong or not connected to a particular subject area.

Therefore, it is needed to complement such ontology learning systems with the appropriate set of ontology optimization procedures. The technique of ontology optimization during its dynamic development is presented in this paper. In works (Montes-y-Gómez et al., 2000) is proved the usefulness of ontology representation in the form of a conceptual graph, vertices of which have certain semantic and numerical characteristics. It is a weighted graph in which at the stage of formation the existence of parallel edges, loops, cycles, duplicate vertices with the same parameters and other features is allowed. Identification and removal of such features for the purpose of ontology graph normalization and the optimization of the graph according to certain criteria taking into consideration the importance weights of edges and vertices must be included in the list of mandatory procedures for ensuring the effective functioning of information systems based on ontologies.

In terms of graph theory the structural optimization of ontology (elimination of conflicts, preservation of the integrity and compliance with restrictions on the maximum volume) is the process of alternation, addition and reduction procedures over the ontology graph within a predetermined range of vertices number while trying to maximize the weighted sum of its vertices and edges. During operation of information system there is a permanent updating of the ontology with new concepts which, in turn, requires periodic decision making regarding the selection of elements for removal (resulting in graph reduction) with preserving integrity of ontology's semantic structure. In this regard there is a need for correction of the structure and content of the ontology.

The following optimality criteria, based on the requirements of ISO/IEC 25000, ISO/IEC/IEEE 29119 and ISO/IEC 25010:2011, for structure and content of the ontology were proposed:

- physical memory occupied by the ontology;
- speed of operation measured as information system's response time to external request (the reaction time to parameters changes in the external environment to which system is sensitive);
- completeness of the ontology, which can be determined using the average percentage of non-trivial (non-zero) responses to requests to it;
- integrity of the ontology, that is the absence of duplications and mutually exclusive axioms; and
- degree of balanced representation of subject area, expressed as a uniform representation of its individual units in the ontology.

Obviously, in order to optimize the ontology data bases in applied information system it is needed to choose not one criterion, but their combination, and the choice of the method of this combination has to be done based on experiences, coming from actual system requirements. To a large extent those criteria are heuristic, they cannot be substantiated by some common to all systems mathematical approach.

Integrity of the knowledge base ontology

The ability of information system to make well-grounded decisions, give answers to the questions asked by the developers or users, requires the presence of the ontology that enables the justification of such decisions (Kanishcheva et al., 2017; Lytvyn et al., 2016; Lytvyn et al., 2018; Vysotska et al., 2018; Naum et al., 2017). In particular, the keeping of consistency in conclusions, giving the same answers to the similar but differently formed questions. Such ontology must satisfy the requirements of integrity. The concept of integrity combines characteristics or requirements, including:

- controlled redundancy;
- connectivity of the ontology graph;
- absence of mutually exclusive claims.

In systems based on knowledge, a redundancy may arise as a result of ontology population, which is manifested by the presence of duplicated structures: concepts and assertions. The coherency of ontology graph is a property, which means that between any two vertices of such graph there is a simple circuit. The connectivity shows that all elements of knowledge base are situated within reach of information system and can be found in process of creating a response to some request. During the reordering and the reduction of the ontology, the system must monitor the condition of its graph coherence and avoid the operations that violate this condition.

The verification of graph's coherence can be done using the results of theorem about the estimation of edges number through the number of vertices and the number of coherence components (Biggs, 1986). If we denote *p* and *q* as the number of vertices and edges in the graph respectively, then the following both conditions must be satisfied:

1. if $q > (p-1)(p-2)/2$, then graph is coherent;
2. in the coherent graph $p-1 \le q \le p(p-1)/2$.

While making changes (adding new elements to the ontology, modification, removal of elements) the system must check its integrity, that is the absence of duplicated and/or mutually exclusive axioms. This can be implemented using the mechanism of identifying the opposite reviews and comparing them while doing the consecutive logical inversion of one of opposing review's axioms by the method of resolutions (Bondy & Murty, 2008). In the case of coincidence of direct and inverted predicates, the system receives a signal of integrity violation and a need to remove the contradictions of statements. In case of mutually exclusive axioms detection, the conflict is solved by removing the one with lower weight.

Limitations of physical memory

The system must be implemented on the basis of specific hardware and software for which there is a limit on the amount of RAM (Random Access Memory). On the other hand, excessive growth of database slows its performance which may be crucial in the case of systems operating in real time mode. During the initial formation of the information system's ontology at the time of its creation such a problem does not occur. However, during the system's operation, the allocated physical memory is filled to full capacity, so it is needed to use the procedures of removing the part of ontology, which is the least efficiently used. Thus, the system operates alternately in two modes: manual and automated learning with two sub-modes each:

1. supplement of ontology by new knowledge data;
2. information removal from the ontology (data which in some sense have the lower value for the user).

One approach to the selection of attributes identifying the knowledge for removal from the ontology, is the weighting of concepts and relations between them during their addition and usage during the system's operation (see: Figure 1).

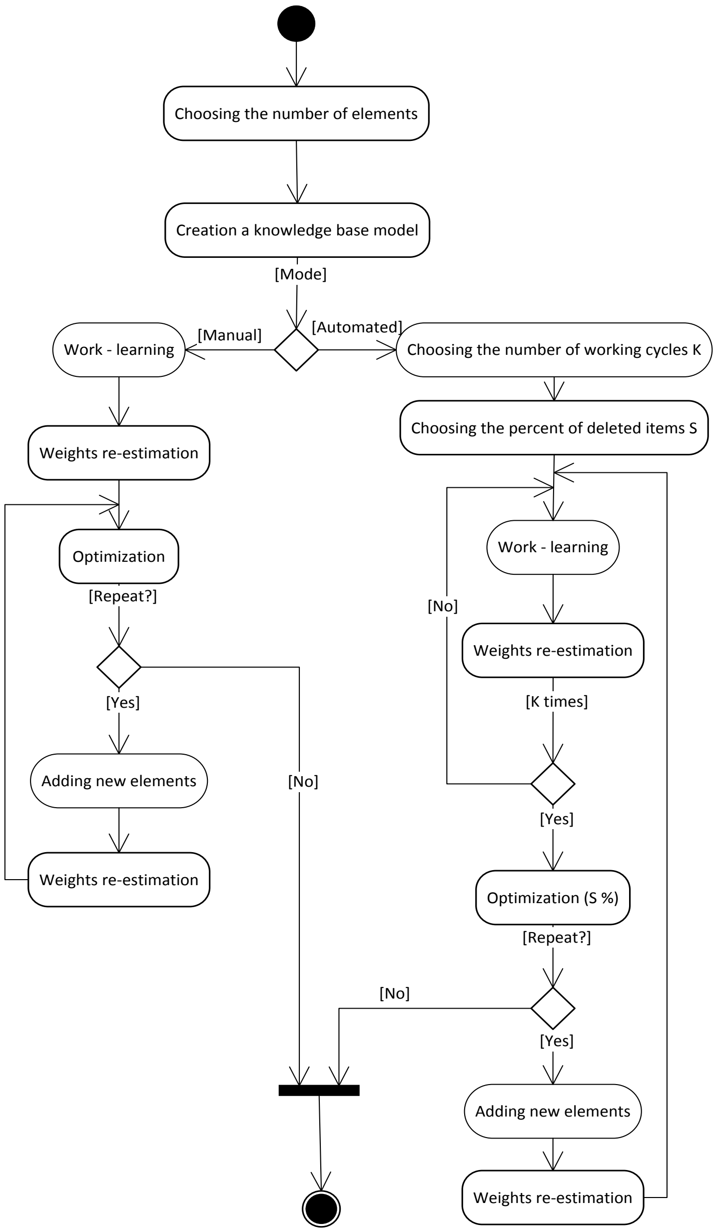

Figure 1. Block-diagram of adaptive ontology learning procedure

To maintain the system in working condition it is necessary to leave a certain amount of free RAM. In this work the threshold of 10 % of the total volume is selected. If at the time of memory allocation this target index is exceeded, the system turns into optimization mode, in which are done the subsequent operations of selection and removal from the ontology of the elements for which the ratio of the importance to occupied memory space is minimal:

$$J_1 = \min_{1 \le j \le K} \frac{W_j}{m_j}, \qquad (1)$$

where W_j is the importance of element C_j; m_j is the place of element C_j in the RAM; K is the number of ontology concepts.

One approach to the selection of attributes identifying the knowledge for removal from the ontology, is the weighting of concepts and relations between them during their addition and usage during the system's operation.

The time of response to external query

The information system's performance, which is determined by the time of response to external query, can be estimated by the maximum number of arcs in the ontology graph for the possible trajectory of message spreading between the concepts which are involved in response generation. Simple and effective measure of the system's performance is provided by the eccentricity of the graph ontology vertices. The eccentricity E_j of a vertex C_j in coherent graph G is the maximum distance from vertex C_j to other vertices in the graph G. Then the worst system performance is estimated as:

$$J_2 = \max_{1 \le j \le K} E_j, \qquad (2)$$

where E_j is eccentricity of vertex C_j in coherent graph G, K is number of ontology concepts.

The completeness of the ontology system

The completeness of the ontology can be defined as the average percentage of non-trivial responses to external queries to the system. The trivial answers are the answers which don't give new information to its recipient. In particular, such answer as "no information available" is trivial. The percentage evaluation of non-trivial responses from certain system can be made by other complete information system, or human expert. The principle of determining the completeness of the ontology is based on the methodology of comparison and evaluation of search engines proposed by the American Institute of standards (NIST) which is the one of the most authoritative bodies of information technology in the United States. This technique uses a corpus of test questions and documents accumulated during the conferences on the assessment of text search systems (TREC – text retrieval evaluation conference), that are made by NIST.

This criterion is integral and applicable only to evaluation and comparison of information systems in general. Therefore, it cannot be used for systematic optimization of the ontology's structure.

Balanced domain representation

During automated ontology learning some ontology classes is much more detailed than other. A balanced domain representation is manifested as a uniform presentation of each separate section in the ontology. The requirement of balanced representation can be applied to meta-ontologies, the scope of which is impossible to determine a priori.

The formal criterion of balanced representation of the concept-class in the ontology can be the variance of the importance of its subclasses:

$$\sigma_j^{<k>} = \overline{\left| \overline{W_i^{<k+1>}} - W_i^{<k+1>} \right|^2}, \ i = \overline{1,n}, \ j = \overline{1,K}, \qquad (3)$$

where $\overline{W_i^{<k+1>}}$, $W_i^{<k+1>}$ is the importance and average importance of level subclasses, respectively; n is number of subclasses in j-th class, N is number of ontology classes.

During optimization of ontology the balancing criteria can be applied in process of selection of sample texts used for ontology learning.

Formulation of the ontology optimization problem

Let us formulate the optimization problem, which will be the basis for the procedures of ontology learning and optimization. Automated generating of ontology determines the need to perform its local optimization during the population and global one at the stage of ordering, when the population is paused until the completion of the optimization procedure. A method of ontology optimization includes the tasks of its structure normalization and content optimization, besides:

1. the removal of parallel edges, duplicated vertices with identical parameters and other features of the ontology graph structure, which can compromise its integrity and reduce the operation efficiency, is the task of normalization of the ontology's structure;
2. the optimization of the ontology content is done in order to increase the system's performance and informational density for the given constraints for the physical memory.

It is assumed that solution of these tasks should be separated, moreover, in order to preserve the integrity of the ontology, a structural validation should be done first, and then the optimization of its content by the sequential reduction of its graph until the requirements using the selected criteria will be met. This is done by maximizing the importance sum of vertices and edges of this graph.

The task of structure optimization

The criteria defined above ensure the integrity of ontology – connectivity, consistency (the absence of mutually exclusive axioms) and minimality (controlled redundancy) and they are the basis of the optimization problem of ISO/IEC 25000, ISO/IEC/IEEE 29119 and ISO/IEC 25010:2011. One of the criteria – the connectivity – will act as a constraint that must be met, but the criteria of minimality and coherence define two separate optimization problems. The problem of minimization of the ontology graph structure is based on a typical optimization problem of graph theory about finding a minimum core, which consists in finding a minimum importance core in a weighted graph. But the task of ensuring coherence in the structure of the graph is effectively solved by employing a method of resolutions. The order of such optimization procedures application is important, because the minimization of the structure is partly achieved during the elimination of logical inconsistencies. So, let's examine them in the order of application.

Ensuring the consistency of ontology structure with method of resolutions

The existence of mutually exclusive claims in the structure of the ontology leads to internal logical conflicts, disrupting its integrity, so the system must have the ability to promptly identify and remove those that are recognized as false. This problem can be effectively resolved by using the resolution method, one of the classical methods of automatic theorem proving (Robinson, 1965). This method is based on the proof by contradiction of consistency for a set of statements. For these purpose formulas of predicate calculus with the help of appropriate transformations are brought to disjunctive form, presented as a set of sentences, belonging to a certain class of well-formed formulas, being a disjunction of literals without quantifiers, each of which is the predicate or the negation of predicate.

The "blind" search as the direct application of resolution method can be considered as disadvantageous because it uses a sorting procedure. This is inefficient, because method generates for large ontologies a significant amount of fruitless resolutions. In order to prevent the "exponential explosion" it is necessary to apply more effective modifications of the method: a semantic resolution, lock resolution, linear resolution. The nature of these modifications is the introduction of certain criteria, according to which disjunctions which should take part in a regular resolution are selected. These modifications generally do not remove backtrack sorting, but can significantly reduce this sorting and make resolutions method totally suitable for practical application. In this work the algorithm of linear resolution is applied taking into consideration its effectiveness and ease of implementation (Bondy & Murty, 2008).

Structure optimization by searching the minimum frame

The foremost purpose of the ontology structure optimization is to ensure its minimality, the lack of information redundancy. In the case of representation of the ontology structure as a weighted graph, where the weights of edges reflect the importance of represented connections and redundancy can manifest in the form of parallel edges, the procedure of eliminating such redundancy may be implemented as successive deletion of edges with minimal weight, while maintaining the connectivity of the entire graph. In order to solve this problem it is proposed to use the algorithm for building a minimum graph frame (Russell & Norvig, 2009). The frame is a subgraph of certain graph $G(V, E)$, containing all its vertices and is a tree. When the edges of the graph are weighted, it is possible to construct a minimal frame tree (MST – Minimal Spanning Tree) of a weighted graph, it is a frame tree, the weight of which (the sum of its edges weights) is not greater than the weight of any other frame tree. In our case, instead of allocating edges with minimal weight, it is needed to select the edges with the maximal weight, taking into consideration the presentational characteristics of a concept graph of a semantic network with different types of semantic relations.

The whole graph of database ontology, as a graph of a semantic network, consists of sub graphs with common vertices (representing concepts) but different types of edges which represent various semantic relations between these concepts. The procedure of constructing a minimal frame tree is performed separately, for each subgraph selected according to the type of semantic relations. So, the general strategy of constructing the minimal frame in the minimization graph structure problem is following: at each step to fragment of frame tree generated in the previous step, is added the edge with maximum weight selected among those edges which are already connected to a built frame tree, but have vertices not yet added. With the purpose of the effective implementation of this process steps, it is needed to associate with each vertex $v \in V$ two labels $\beta(v)$ and $\gamma(v)$, representing respectively a weight of edge with maximum weight connecting a vertex v with built fragment of maximum frame and the name of the second vertex of this edge. The labels $\beta(v)$ and $\gamma(v)$ provide the ability to quickly find at each step an edge with maximum weight.

It is assumed that the graph G which reflects the structure of ontology of some semantic type is specified by the matrix of weights Q of weighted conceptual graph. Then the algorithm of the minimum frame building will be described as below (See Figure 2).

1. Choose any graph vertex of database graph G and find the neighbour vertex with the greatest weight of edge. In this case $ET = \varnothing$, $VT = \{a\}$, where VT, ET are the

sets of vertices and edges of minimal frame fragment that is built; *a* is any vertex *G*.

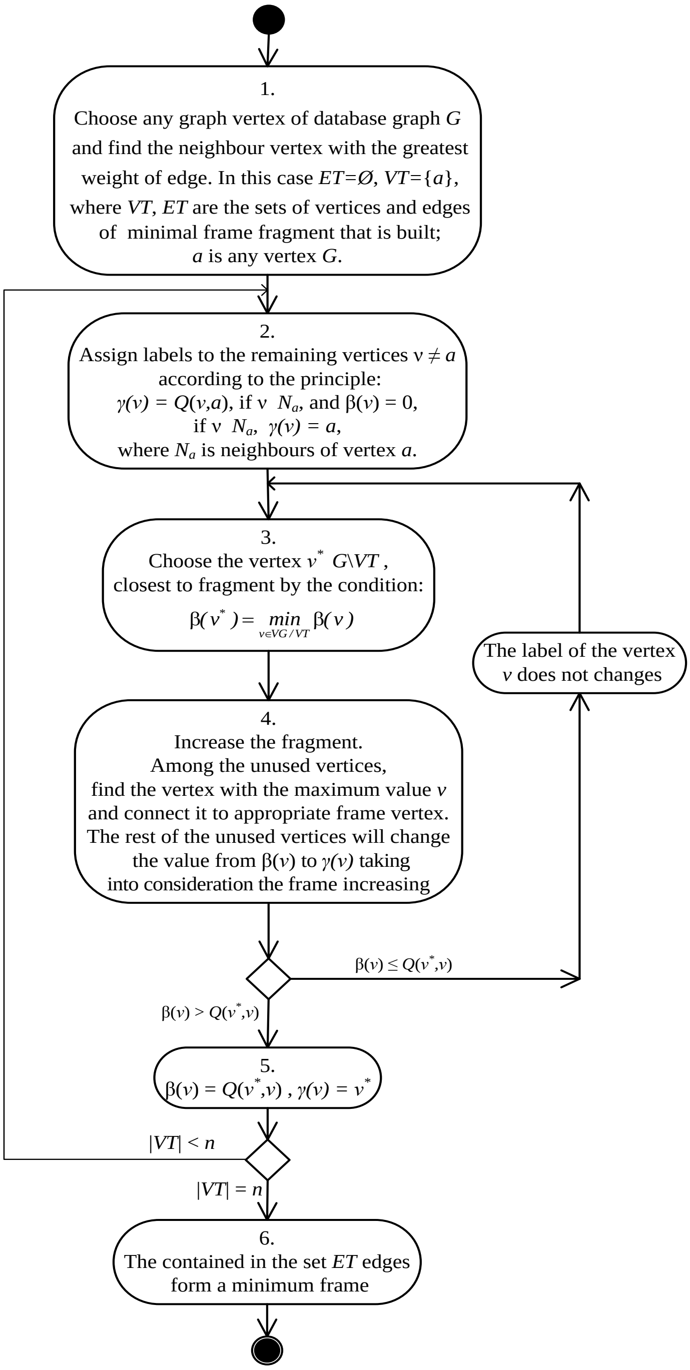

Figure 2. The algorithm of the minimum frame building

2. Assign labels to the remaining vertices $v \neq a$ according to the principle: $\gamma(v) = Q(v,a)$, if $v \in N_a$, and $\beta(v) = 0$, if $v \notin N_a$, $\gamma(v) = a$, where N_a is neighbours of vertex *a*.
3. Choose the vertex $v^* \in G \setminus VT$, closest to fragment by the condition:

$$\beta(v^*) = \min_{v \in VG/VT} \beta(v). \qquad (4)$$

4. Increase the fragment. Among the unused vertices, find the vertex with the maximum value *v* and connect it to appropriate frame vertex. The rest of the unused vertices will change the value from $\beta(v)$ to $\gamma(v)$ taking into consideration the frame increasing, it means if $\beta(v) > Q(v^*, v)$, then $\beta(v) = Q(v^*, v)$, $\gamma(v) = v^*$. If $\beta(v) \leq Q(v^*, v)$, then the label of the vertex *v* does not changes. Then move to step 3.
5. The operations are repeated until there is no unused vertices, if $|VT| = n$, the procedure must be terminated. The contained in the set *ET* edges form a minimum frame.

In Figure 3 the process of creation of minimum frame after each iteration of the algorithm is shown.

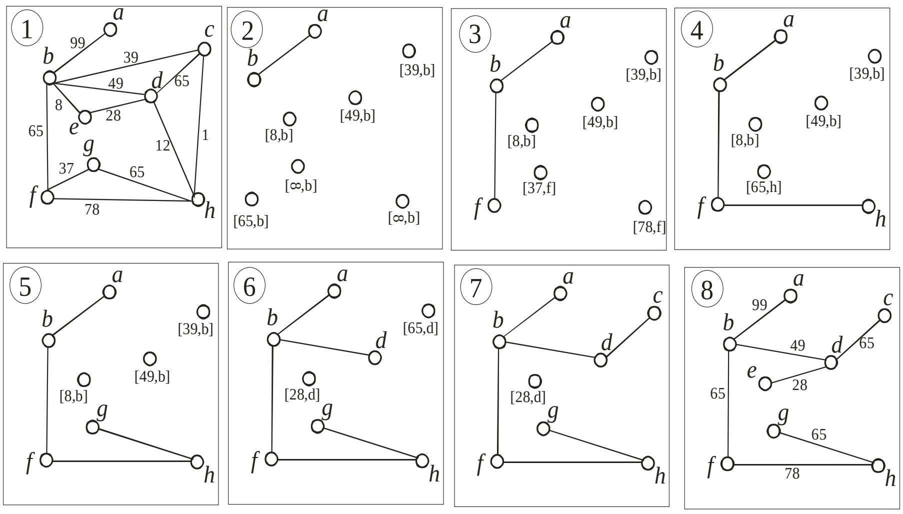

Figure 3. A graph minimum frame creation example

The graph $T = (VT, ET)$ at each step remains to be a tree, because every time the edge is added to *ET*, one end of which belongs to *VT*, and the other doesn't. After the completion of the algorithm a tree is a frame, because the algorithm stops when *VT*=*G*. Single completion of step 1 requires $O(|V|)$ time. The same time is necessary to update the labels in the step 4, and the increase of fragment is made within time *O*(1). Since each of 2-5 steps is executed *n*-1 times, then the estimation of the complexity of the algorithm is $O(|V|^2)$.

The problem of ontology content optimization

Criteria were selected and justified according to which ontology content optimality is achieved: the performance, completeness and minimum amount of physical memory. The optimization procedure of the ontology implies the successive removal of concepts that have the least importance. At this stage, based on specified criteria, it is necessary to formulate the optimization problem, to build a target function and to impose appropriate restrictions arising from the implementation of information system. Since the optimality of the ontology content of the system is evaluated by conflicting criteria, then "to bring them to a common denominator" at the same time is impossible. In this case we have a situation of uncertainty of objectives, which can be eliminated by using one of the methods of uncertainty elimination (control indicators, the linear convolution method, the introduction of a metric in the space of objective functions, Pareto compromise, binarization and removing binary artifacts) (Kravets 2010; Ramı et al., 2002; Precup et al., 2013; Ramirez-Ortegon et al., 2013).

So, firstly it is needed to solve the problem of finding the optimal number of concepts in the ontology and then, by using the obtained results, to implement a procedure of periodic alternating updating of the ontology with new concepts and removal from the ontology the elements with the least semantic value for the system. This procedure should be the basis of a method of ontology adaptation to the domain specified by the user as a functional element of the corresponding information system. The task of removal of the least important elements from ontology can be formulated as an optimization problem about the backpack (Poirriez et al., 2009). Here a backpack will represent a part of ontology's concepts, which are subject to removal. It is needed to consider the constraints that are imposed to such optimization problem, to define the optimal size of ontology according to criteria of performance/completeness and to formalize in the form of the algorithm the solving of the problem of elements selection for removal from the ontology. For practical applications of discrete optimization problem about the backpack instead of the integral optimality criteria of the ontology is generally necessary to apply the criteria that characterize each individual element of the ontology and provide an opportunity to make a decision about whether or not to add it to the ontology (or remove from it). It is assumed that each element of the ontology occupies the same amount of RAM, and therefore, the total occupied volume is linearly dependent on the number of concepts specified in the ontology. The average eccentricity of the vertices in the ontology graph is taken into account while defining the self-weight of classes and weight of their subclasses, going down by the level. Taking into account the fact that the ontology contains more than 10-100 thousands of individual concepts, arranged in taxonomic of 15-25-level hierarchy, the difference between the weight of the concepts close to the root and the weight of the concepts of lower levels is several orders of magnitude.

Discussion

By applying a simple sorting procedure for ontologies with n elements, an upper assessment of the number of possible combinations is 2^n. In order to resolve this task it is enough to apply the methods of dynamic programming, greedy algorithm or method of branches and borders. During the effectiveness study of the individual methods of implementation, the greedy algorithm was selected. This algorithm has complexity $O(n \cdot \log n)$ and guarantees finding a solution not worse than twice from the optimal one, which is acceptable for this task. So, an ontology with n elements has the over volume $N = M/10$, where M is the maximum allowable amount. If entering of the i-th element to “backpack” x_i, $0 \le x_i < 1$ determines benefit x_i/W_i, the optimal filling of the “backpack” is such a filling of elements that maximizes the total profit. The task can be formulated in the following way: maximize $\sum_{i=1}^{n} \frac{1}{W_i} x_i$ of such elements for which $W_i > 0$, $m_i > 0$, $i = \overline{1,n}$:

$$\sum_{i=1}^{n} m_i x_i \le N \,, \quad \begin{cases} x_i = 0, if \;\; is, \\ x_i = 1, if \;\; not \; is. \end{cases} \qquad (5)$$

The problem is simplified if we assume $m_i = m = const$ that generally corresponds to the conditions of implementation of the ontology. Implementation of the greedy algorithm in this task is reduced to the successive execution of the following operations:

1. find the element with the lowest importance W_i;
2. add it to the set of those that are withdrawn (assignment x_i =1);
3. calculate and check the conditions (4);
4. if condition (4) is met go to step 1 of the algorithm;
5. in case of violation of the conditions stop the search and remove the selected elements.

Execution time assessment for the algorithm depends entirely on the time used by sorting algorithm, because for the implementation of the population strategy after elements ordering the time $O(n)$ is needed. Taking into consideration the reasons of efficiency for the whole system, it is not appropriate to apply the optimization procedure every time, but periodically when certain conditions are fulfilled (such as the time limit or the volume available for population is exceeded), alternating stages of ontology population and optimization. The criteria that need to be selected to determine the moments of transition from mode to mode need to be investigated based on the results of the numerical modelling. So, dynamic ontology population and periodic optimization of its structure and content provide restructuring and improvement of the domain model, represented by its ontology.

Conclusions

Automated ontology population (an ontology learning) causes the problem of redundancy and non-optimality that causes necessity of its periodic optimization. It is justified a set of optimality criteria for an ontology structure and content that reflect the requirements for the operating characteristics and the technical feasibility of intelligent systems. The problem of ontology optimization is proposed to divide into the optimization of its structure and content. The task of ontology structure optimization consists in elimination parallel edges, cycles, loops, and logical contradictions that disrupt its integrity and decrease the efficiency of the operation. The resolution method and the algorithm to find the minimal spanning tree were applied to solve this problem. Optimization of the ontology content is conducted to increase its information richness and to ensure its adaptation to the specific domain. This optimization task can be reduced to the backpack task the effective solving algorithms for which are known. The application of optimization techniques provides controlled automated ontology learning that significantly expands the usability of such systems and reduces the time expenditures on their implementation.

Acknowledgements

The article describes the results or research performed in Information systems and networks department in Lviv Polytechnic National University. We are grateful to our colleagues for their support and understanding. All authors have completed the Unified Competing Interest form (available on request from the corresponding author) and declare: no support from any organization for the submitted work; no financial relationships with any organizations that might have an interest in the submitted work in the previous 3 years; no other relationships or activities that could appear to have influenced the submitted work.

References

Abolhassani, H., Bagheri-Hariri, B., & Haeri, S.H. (2006). On ontology alignment experiments. *Webology*, 3(3), Article 28. Retrieved July 15, 2018, from http://www.webology.org/2006/v3n3/a28.html

Ahmad, M.N., & Colomb, R.M. (2007). Overview of ontology servers research. *Webology*, 4(2), Article 43. Retrieved July 15, 2018, from http://www.webology.org/2007/v4n2/a43.html

Alipanah, N., Parveen, P., Khan, L., & Thuraisingham, B. (2011). Ontology-driven query expansion using map/reduce framework to facilitate federated queries. *Int. Conf. on Web Services*, 712-713.

Basyuk, T. (2015). The main reasons of attendance falling of internet resource. *Computer Science and Information Technologies*, 91-93.

Biggs, N., Lloyd, E., & Wilson, R. (1986). *Graph theory*. Oxford: Oxford University Press.

Bondy, J.A., & Murty, U.S.R. (2008). *Graph theory*. London: Springer-Verlag.

Calvaneze, D. (2013). *Optimizing ontology-based data access*. KRDB Research Centre for Knowledge and Data. Bozen-Bolzano, Italy.

Chen, J., Dosyn, D., Lytvyn, & V., Sachenko, A. (2016). Smart Data Integration by Goal Driven Ontology Learning, *Advances in Big Data*, pp. 283-292.

Davydov, M., & Lozynska, O. (2017). Information System for Translation into Ukrainian Sign Language on Mobile Devices. *International Conference on Computer Science and Information Technologies*, 48-51.

Davydov, M., & Lozynska, O. (2017). Linguistic Models of Assistive Computer Technologies for Cognition and Communication. *International Conference* on *Computer Science and Information Technologies*, 171-175.

Euzenat, J., & Shvaiko, P. (2007). Ontology Matching. *Springer*, Heidelberg, Germany.

Fedushko, S. (2014). Development of a software for computer-linguistic verification of socio-demographic profile of web-community member. *Webology*, 11(2), Article 126. Retrieved July 15, 2018, from http://www.webology.org/2014/v11n2/a126.pdf

Gottlob, G., Orsi, G., & Pieris, A. (2011). Ontological queries: Rewriting and optimization. In *Proceedings of the 27th International Conference on Data Engineering, ICDE 2011*, April 11-16, 2011, Hannover, Germany (pp. 2-13). IEEE. DOI: 10.1109/ICDE.2011.5767965

Gozhyj, A., Chyrun, L., Kowalska-Styczen, A., & Lozynska, O. (2018). Uniform Method of Operative Content Management in Web Systems. *CEUR Workshop Proceedings*, 2136, pp. 62-77.

ISO/IEC 25010:2011 - Systems and software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE) - System and software quality models. Retrieved July 15, 2018, from http://www.iso.org/iso/catalogue_detail.htm?csnumber=35733.

ISO/IEC/IEEE 29119 Software Testing. Retrieved July 15, 2018, from http://www.softwaretestingstandard.org/

ISO/IEC 25000 series of standards. Retrieved July 15, 2018, from http://iso25000.com/index.php/en/iso-25000-standards.

Kanishcheva O., Vysotska, V., Chyrun, L., & Gozhyj, A. (2017). Method of Integration and Content Management of the Information Resources Network, *Advances in Intelligent Systems and Computing*, 689, 204-216.

Keet, C.M., Ławrynowicz, A., d'Amato, C., & Hilario, M. (2013). Modeling issues and choices in the data mining optimization ontology.

Keet, C.M., Ławrynowicz, A., d'Amato, C., Kalousis, A., Nguyen, P., Palma, R., Stevens, R., & Hilario, M. (2015). The data mining optimization ontology. *Web Semantics: Science, Services and Agents on the World Wide Web*, 32, 43-53.

Khomytska, I., & Teslyuk, V. (2016). Specifics of phonostatistical structure of the scientific style in English style system. *Computer Science and Information Technologies*, 129-131.

Khomytska, I., & Teslyuk, V. (2017). The Method of Statistical Analysis of the Scientific, Colloquial, Belles-Lettres and Newspaper Styles on the Phonological Level. *Advances in Intelligent Systems and Computing*, 512, 149-163.

Korobchinsky, M., Vysotska, V., Chyrun, L., & Chyrun, L. (2017). Peculiarities of content forming and analysis in internet newspaper covering music news. *International Conference on Computer Science and Information Technologies*, 52-57.

Korzh, R., Peleschyshyn, A., Syerov, Y., & Fedushko, S. (2014). The cataloging of virtual communities of educational thematic. *Webology*, 11(1), Article 117. Retrieved July 15, 2018, from http://www.webology.org/2014/v11n1/a117.pdf

Kravets, P. (2010). The control agent with fuzzy logic. *Perspective Technologies and Methods in MEMS Design*, pp. 40-41.

Li, Y., & Heflin, J. (2010). Query optimization for ontology-based information integration. *Proceedings of the 19th ACM Conference on Information and Knowledge Management, CIKM 2010*, Toronto, Ontario, Canada, October 26-30, 2010, pp. 1369-1372.

Lytvyn, V., Sharonova, N., Hamon, T., Vysotska, V., Grabar, & N., Kowalska-Styczen, A. (2018). Computational linguistics and intelligent systems. *CEUR Workshop Proceedings*, Vol. 2136.

Lytvyn, V., Vysotska, V., Chyrun, L., & Chyrun, L. (2016). Distance Learning Method for Modern Youth Promotion and Involvement in Independent Scientific Researches. *International Conference on on Data Stream Mining & Processing* (DSMP), 269-274.

Lytvyn, V., Vysotska, V., Peleshchak, I., Rishnyak, I., & Peleshchak, R. (2018). Time dependence of the output signal morphology for nonlinear oscillator neuron based on Van der Pol Model. *International Journal of Intelligent Systems and Applications*, 10, 8-17.

Lytvyn, V., Vysotska, V., Pukach, P., Bobyk, I., & Pakholok, B. (2016). A method for constructing recruitment rules based on the analysis of a specialist's competences. *Eastern-European Journal of Enterprise Technologies*, 6/2(84), 4-14.

Lytvyn, V., Vysotska, V., Uhryn, D., Hrendus, M., & Naum, O. (2018). Analysis of statistical methods for stable combinations determination of keywords identification. *Eastern-European Journal of Enterprise Technologies*, 2/2(92), 23-37.

Lytvyn, V., & Vysotska, V. (2015). Designing architecture of electronic content commerce system. In: Computer Science and Information Technologies. *International Conference on Computer Science and Information Technologies*, 115-119.

Lytvyn V., Vysotska, V., Pukach, P., Bobyk, I., & Uhryn, D. (2017). Development of a method for the recognition of author’s style in the Ukrainian language texts based on linguometry, stylemetry and glottochronology. *Eastern-European Journal of Enterprise Technologies*, 4/2(88), 10-18.

Lytvyn, V., Pukach, P., Bobyk, I., & Vysotska, V. (2016). The method of formation of the status of personality understanding based on the content analysis. *Eastern-European Journal of Enterprise Technologies*, 5/2(83), 4-12.

Lytvyn, V., Vysotska, V., Burov, Y., Veres, O., & Rishnyak, I. (2018). The Contextual Search Method Based on Domain Thesaurus. *Advances in Intelligent Systems and Computing*, 689, 310-319.

Lytvyn, V., Vysotska, V., Veres, O., Rishnyak, I., & Rishnyak, H. (2017). Classification Methods of Text Documents Using Ontology Based Approach. *Advances in Intelligent Systems and Computing*, 512, 229-240.

Lytvyn, V.,Vysotska, V.,Pukach, P.,Brodyak, O., & Ugryn, D. (2017). Development of a method for determining the keywords in the slavic language texts based on the technology of web mining. *EasternEuropean Journal of Enterprise Technologies*, 2/2(86), 4-12.

Maedche A., & Staab S. (2002). Measuring similarity between ontologies. In: Gómez-Pérez A., Benjamins V.R. (eds.), *Knowledge Engineering and Knowledge Management: Ontologies and the Semantic Web. EKAW 2002*. Lecture Notes in Computer Science, vol 2473. Springer, Berlin, Heidelberg, pp. 251-263.

Maksymiv, O., Rak, T., & Peleshko, D. (2017). Video-based Flame Detection using LBP-based Descriptor: Influences of Classifiers Variety on Detection Efficiency. *International Journal of Intelligent Systems and Applications*, 9(2), 42-48.

Martin, D., Toro, del R., Haber, R., & Dorronsoro, J. (2009). Optimal tuning of a networked linear controller using a multi-objective genetic algorithm and its application to one complex electromechanical process. *International Journal of Innovative Computing, Information and Control*, 5/10(B), 3405-3414.

Martinez-Gil, J., Alba, E., & Aldana-Montes, J.F. (2008). Optimizing ontology alignments by using genetic algorithms. *The workshop on nature based reasoning for the semantic Web*. Karlsruhe, Germany.

Montes-y-Gómez, M., Gelbukh, A., & López-López, A. (2000). Comparison of Conceptual Graphs. *Artificial Intelligence*, 1793. Retrieved July 15, 2018, from https://pdfs.semanticscholar.org/ 572a/8355404a7e6e909a9e923f5d469a cb9ec347.pdf.

Naum, O., Chyrun, L., Kanishcheva, O., & Vysotska, V. (2017). Intellectual system design for content formation. *Computer Science and Information Technologies*, 131-138.

Peleshko, D., Rak, T., & Izonin, I. (2016). Image superresolution via divergence matrix and automatic detection of crossover. *International Journal of Intelligent Systems and Application*, 8(12), 1-8.

Poirriez, V., Yanev, N., & Andonov, R. (2009). A hybrid algorithm for the unbounded knapsack problem. *Discrete Optimization*, 6(1),110-124.

Precup, R.-E., David, R.-C., Petriu, E.M., Preitl, S., & Rădac, M.-B. (2013). Fuzzy logic-based adaptive gravitational search algorithm for optimal tuning of fuzzy controlled servo systems. *IET Control Theory & Applications*, 7(1), 99-107.

Rami, J., & Vlach, M. (2002). Pareto-optimality of compromise decisions Fuzzy Sets and Systems, 129(1), 119-127.

Ramirez-Ortegon, M. A., Margner, V., Cuevas, E., & Rojas, R. (2013). An optimization for binarization methods by removing binary artifacts. *Pattern Recognition Letters*, 34(11), 1299-1306.

Rashkevych, Y., Peleshko, D., Vynokurova, O., Izonin, I., & Lotoshynska, N. (2017). Single-frame image super-resolution based on singular square matrix operator. *2017 IEEE First Ukraine Conference on Electrical and Computer Engineering* (UKRCON), Ukraine.

Robinson, J. Alan. (1965). A machine-oriented logic based on the resolution principle. *Journal of the ACM* (JACM), 12(1), 23-41.

Russell, S.J., & Norvig, P. (2009). *Artificial intelligence: A modern approach*. (3^{rd} ed.). Prentice Hall.

Rusyn, B., Lutsyk, O., Lysak, O., Lukeniuk, A., & Pohreliuk, L. (2016). Lossless Image Compression in the Remote Sensing Applications. *International Conference on Data Stream Mining & Processing* (DSMP), pp. 195-198.

Solos, I. P., Tassopoulos, I. X., & Beligiannis, G. N. (2016). Optimizing shift scheduling for tank trucks using an effective stochastic variable neighbourhood approach. *International Journal of Artificial Intelligence*, 14(1), 1-26.

Su, J., Vysotska, V., Sachenko, A., Lytvyn, V., & Burov, Y. (2017). Information resources processing using linguistic analysis of textual content. *Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications*, 573-578.

Tkachenko, R., Tkachenko, P., Izonin, I., & Tsymbal, Y. (2018). Learning-based image scaling using neural-like structure of geometric transformation paradigm. *Studies in Computational Intelligence*, 730, 537–565.

Vysotska, V., Hasko, R., & Kuchkovskiy, V. (2015). Process analysis in electronic content commerce system. *International Conference Computer Sciences and Information Technologies*, 120-123.

Vysotska, V., Fernandes, V.B., & Emmerich, M. (2018). Web content support method in electronic business systems. *CEUR Workshop Proceedings*, 2136, 20-41.

Vysotska, V., Lytvyn, V., Burov, Y., Gozhyj, A., & Makara, S. (2018). The consolidated information web-resource about pharmacy networks in city. *CEUR Workshop Proceedings*, 2255, 239-255.

Xue, X., Wang, Y., & Hao, W. (2015). Optimizing ontology alignments by using NSGA-II. *The International Arab Journal of Information Technology*, 12(2), 176-182.

Zhezhnych P., & Markiv O. (2018) Linguistic Comparison Quality Evaluation of Web-Site Content with Tourism Documentation Objects. In: Shakhovska N., Stepashko V. (eds), *Advances in Intelligent Systems and Computing II. CSIT 2017. Advances in Intelligent Systems and Computing*, Vol. 689. Springer, Cham, pp. 656-667.

Bibliographic information of this paper for citing:

Lytvyn, Vasyl, Vysotska, Victoria, Dosyn, Dmytro, & Burov, Yevhen (2018). "Method for Ontology content and structure optimization, provided by a weighted conceptual graph." Webology, 15(2), Article 173. Available at: http://www.webology.org/2018/v15n2/a173.pdf

---