Proceedings of the International Multiconference on Computer Science and Information Technology pp. 33–39
ISSN 1896-7094

Creation and Application of Maps of Concepts for Description Logics Ontologies

Krzysztof Goczyła, Teresa Zawadzka, Wojciech Waloszek, Michał Zawadzki

Gdańsk University of Technology, Department of Software Engineering,
ul. Gabriela Narutowicza 11/12, 80-952 Gdańsk, Poland
{kris,tegra,wowal,michawa}@eti.pg.gda.pl

Abstract. In previous work a novel knowledge representation, called Knowledge Cartography, was introduced. The method allows for description, in the form of a map of concepts, of interrelationships among concepts distinguished in a terminology and for gradual (with growth of our knowledge) assignment of individual objects to those concepts. Effectiveness of the process of building map of concepts is a key factor influencing usability of the method. This paper presents a new map-creating algorithm *TreeFusion* which exploits binary decision diagrams originally developed for supporting VLSI design. The paper presents also some current applications of Knowledge Cartography.

1 Introduction

In previous papers [1, ?] we presented a new method of representation of knowledge formulated in terms of Description Logics called *Cartographic Approach*, or *Knowledge Cartography*. By knowledge representation we mean here a mapping from domain of DL terms into another domain, and vice versa. For a given representation to be sensible, two requirements must be met: the mapping must be performed in acceptable time and the other (target) representation (a result of the mapping) should allow for easier realization of some important tasks, such as standard and non-standard inference tasks.

This paper addresses both issues in the context of Cartographic Approach. The *map of concepts* is the main notion in the approach. A map of concepts describes how (possibly complex) DL concepts are mapped into another domain: the domain of binary signatures. Specifically, this paper focuses on an efficient algorithm of creating a map of concepts. The algorithm, called *TreeFusion*, is considerably faster than another algorithm used previously, and enables a system to load and process huge ontologies. We also present tools built so far that exploit the Cartographic Approach. The tools are able to make use of ease of signature analyses and transformations.

The rest of the paper is organised as follows: In Section 2 we make a brief introduction to Knowledge Cartography. Section 3 gives details of *TreeFusion* algorithm that creates a map of concepts. Section 4 overviews recent applications of the Cartographic Approach. A summary concludes the paper.

2 Knowledge Cartography and maps of concepts

Knowledge Cartography has been introduced to speed up the process of inference, especially about large numbers of individuals. Good results obtained with querying the description of the world came at cost of time-consuming analysis of description of terminology, which has to be performed at initial stage to obtain *the map of concepts*.

Maps of concepts are in fact a concise description of interrelationships among concepts defined in a terminology. At the current stage terminologies expressed in $\mathcal{ALC}$ can be handled by Cartographic Approach [1]. In this section, the following example terminology is used:

$$\begin{aligned} &Woman \equiv Person \sqcap \neg Male \\ &Man \equiv Person \sqcap \neg Woman \\ &Parent \equiv Person \sqcap \exists hasChild.\top \\ &Mother \equiv Parent \sqcap \neg Male \\ &\exists hasChild.Person \equiv \exists hasChild.\top \sqcap Person \end{aligned} \tag{1}$$

The graphical representation of a map of concepts resembles a Venn diagram that shows domains of concepts in a terminology as areas on the map. An important thing is that unsatisfiable areas (i.e. areas to which any individual cannot belong) are removed from the map. The procedure of removing unsatisfiable areas from a map is illustrated in Fig. 1.

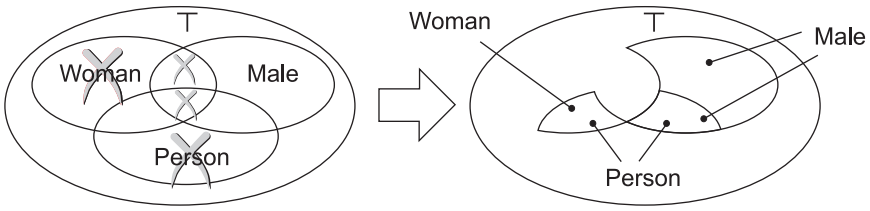
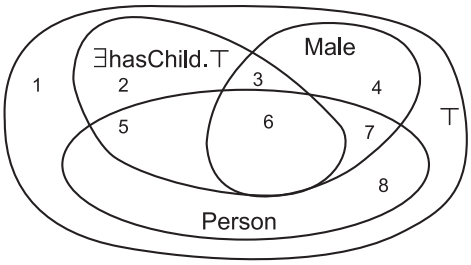


Fig. 1. The first axiom from the terminology (1) results in some areas being removed from the initial map of three concepts: *Woman*, *Male*, and *Person*



Person	00001111
Male	00110110
Parent	00001100
Mother	00001000
Woman	00001001
∃hasChild.⊤	01101100
∃hasChild.Person	00001100

Fig. 2. The graphical and the binary representation of a map of concepts for the terminology (1)

The areas removed after analysis of the first axiom in (1) represent the following concepts: $Woman \sqcap \neg Person \sqcap Male$ (as $Woman \sqsubseteq Person$), $Woman \sqcap \neg Person \sqcap \neg Male$ (for the same reason), $Woman \sqcap Person \sqcap Male$ (as $Woman \sqsubseteq \neg Male$) and $\neg Woman \sqcap Person \sqcap \neg Male$ (as $Person \sqcap \neg Male \sqsubseteq Woman$).

By a *region* we mean an area in a map of concepts that does not contain any other area. A binary representation of the map is created by assigning to regions consecutive natural numbers. In this way, each area in a map of concepts is represented by a string of binary digits (bits) called a *signature*. The length of all signatures equals to the number of regions in a map. A "1" at the k-th position in a signature denotes that the region numbered k is included in the area represented by this signature. Each concept can be assigned such a signature, because any concept is represented by an area in the map. In this way we obtain a set of signatures as a binary representation of a map of concepts, as shown in Fig. 2. (Note that concepts of the form $\exists R.C$ are included in a map only if they are given explicit in a terminology[1].) The signature representation is convenient as many inferences can be performed by executing binary operations, e.g. equivalence of concepts can be determined just by checking equality of signatures ([1],[2]).

In Cartographic Approach the map of concept is created once (it is assumed that the terminology is constant in time) and then used to perform inference over TBox and ABox. As can be read in [1] and [2], the process of creation map of concepts is time consuming but further use of the map allows to obtain shorter response time to queries concerning ontologies with large ABoxes than offered by other reasoners.

In the following we present a *TreeFusion* algorithm that allows to substantially shorten the time to create the map of concepts.

3 The algorithm to create a map of concepts

Currently, for the creation of map of concepts, the *TreeFusion* algorithm is used. An effect of the algorithm is an assignment of signatures to atomic concepts and concepts of the form of $\exists R.C$ appearing in any axiom in terminology (called jointly *cartographic concepts*).

The *TreeFusion* algorithm is based on *Ordered Binary Decision Diagrams* (*OBDD*) [3], [4]. These diagrams have been developed primarily for VLSI circuits design. They allow to design circuits that realize functions with thousands of variables. An OBDD diagram has a form of a binary tree. Each non-terminal vertex v is assigned a natural number denoted $index(v)$. From each non-terminal vertex come out two edges denoted respectively 0 and 1. Vertices to which these edges come are denoted $low(v)$ and $high(v)$, respectively. Leaves are assigned a logical value of 0 or 1.

[1] Concepts in the form of $\forall R.C$ are transformed to the equivalent form of $\neg\exists R.\neg C$ and therefore also represented in the map.

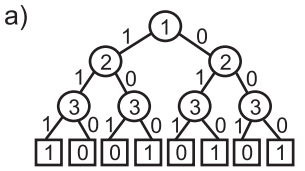


Fig. 3. An example of OBDD diagram before (a) and after (b) reduction

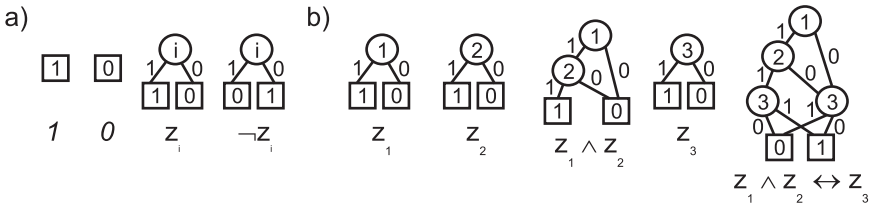


Fig. 4. Elementary trees (a) and steps to create a diagram for the logical formula $z_1 \wedge z_2 \leftrightarrow z_3$ (b)

Each binary diagram can be treated as a tree representing some logical formula. Numbers assigned to non-terminal vertices can be treated as indices of variables used in the formula. We define a *cofactor* of variables for a tree d as a function assigning to indices (and variables) used in the tree d values from $\{0, 1\}$. In this way we can define a value of logical formula for any cofactor by traversing the tree from the root and moving from the current vertex v to $low(v)$ if the variable z_i with index $i = index(v)$ has been assigned value of 0, or to $high(v)$, if 1. The procedure terminates in the leaf—its value is the logical value for the whole formula (if for specified cofactor the value of the formula is 1, such assessment is called *satisfiable*).

The diagram's ordering guarantees that for each vertex v the conditions $index(v) < index(low(v))$ and $index(v) < index(high(v))$ are fulfilled, if only the specified edges lead to non-terminal vertices. An example of an ordered binary diagram is shown in Fig. 3a.

In the *TreeFusion* algorithm we applied structures and processing methods proposed in [3]. The fundamental feature of the diagrams is the fact that they are kept in a *reduced form*. It means that there are no repeating subtrees and the diagrams are turned into form of general digraphs (see Fig. 3b). The reduction of a tree is performed by the *reduce* procedure [3].

Two ordered and reduced decision trees d_1 and d_2 describing formulas f_1 and f_2 respectively can be joined together with respect to some logical operation *op* with the use of *apply* procedure. If the same variables in both formulas have been assigned the same indices, the resulting tree d represents the formula $f_1 op f_2$ [3]. The *apply* procedure takes as parameters the roots of both trees (respectively v_1 and v_2) and the operation *op*. Time complexity of *apply* procedure for graphs G_1 and G_2 is proven to be $O(|G_1||G_2|)$.

By using the *apply* procedure, we are able (using simple trees depicted in Fig. 4a) to build a complex logical expression. Figure 4b describes how to build such a tree.

We exploited OBDD trees in *TreeFusion* as follows. For each axiom a in a terminology T expressed in $\mathcal{ALC}$ a logical formula $f(a)$ is built according to the following rules: $f(C \equiv D) \mapsto f(C) \leftrightarrow f(D)$, $f(C \sqsubseteq D) \mapsto f(C) \rightarrow f(D)$, $f(\neg C) \mapsto \neg f(C)$, $f(C \sqcap D) \mapsto f(C) \wedge f(D)$, $f(C \sqcup D) \mapsto f(C) \vee f(D)$, $f(\forall R.C) \mapsto f(\neg \exists R.\neg C)$, $f(\top) \mapsto 1$, $f(\bot) \mapsto 0$, $f(\exists R.C) \mapsto z_{\exists R.C}$, $f(A) \mapsto z_A$ (A is an atomic concept). Then the formula is turned into a binary decision diagram, under the condition that cartographic concepts are assigned variables with specific indices (there must exists a bijection g for this assignment, see Fig. 5). Using this diagram one can determine signatures of concepts satisfying an axiom being processed. Each cofactor represented by a descending path leading to a leaf with value 1 (*positive path*) also represents one column in a signature, i.e. a single region. During processing of a terminology a tree D is being built. The tree represents a formula which is a conjunction of formulas $f(a)$ for each axiom a in the terminology. The processing of each subsequent axiom triggers the following operation: $D := apply(D, f(a), \wedge)$.

Direct use of OBDD trees allowed for processing large terminologies. However, the algorithm turned out vulnerable to ordering of axioms in a terminology processed. This problem has been solved by the method described below, which turned out also to substantially improve the scalability of the algorithm.

The idea exploited in *TreeFusion* is based on the observation that combining two diagrams with $\wedge$ (AND) operation can be done in $O(1)$ time if ranges of indices of variables in the two trees (ranges from the lowest index to the greatest index used in a tree) are disjoint (we call this operation *join*; see Fig. 6a).

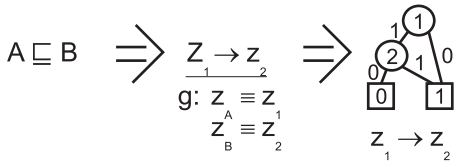


Fig. 5. An example of a process of building a tree for an axiom

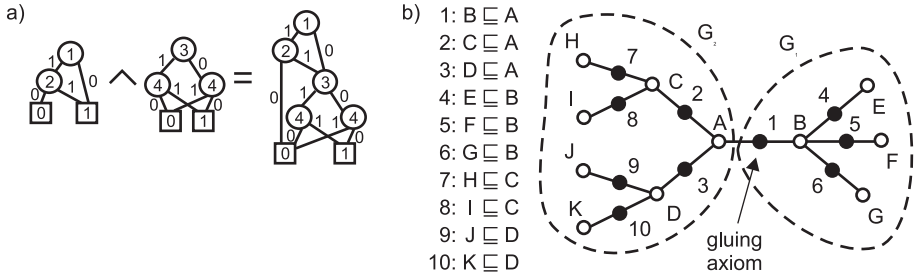


Fig. 6. Illustration of idea of optimizing *TreeFusion*: (a) joining trees with disjoint ranges of indices, (b) graph of axioms and its bisection

Since variables correspond to concepts, the idea was to find groups of axioms referring to disjoint sets of concepts.

This idea was put into work by building a *graph of axioms*. Graph of axioms is a bipartite graph in which two classes of nodes are used: "black" ones representing axioms, and "white" ones representing cartographic concepts. A black node is connected with a white node iff the concept represented by the white node is used in the axiom represented by the black node. To determine groups of axioms referring to disjoint sets of concepts, a *graph bisection* algorithm has been exploited. *Graph bisection* is a problem of finding a smallest set of edges whose removal separates the graph into two components whose sizes (number of nodes) are similar.

This idea is presented in Fig. 6b. There is shown a graph of axioms for an exemplary terminology. Use of bisection separated the graph into two components. The variables for concepts have been indexed in such a way that ranges of indices for the two components are disjoint. Trees for both parts of the terminology will be built independently and then joined. The joint tree D will be then combined with $\wedge$ (AND) operation with a tree for the *gluing axiom*, i.e. axiom represented by the node incident to the separating edge.

This indexing scheme allows for substantial reduction of execution time. The two trees are built independently and then joint by a $O(1)$ operation. The gain obtained in this way has been illustrated in Fig. 7: execution time without optimization is proportional to the area greyed in Fig. 7a. In this figure we assume that the size of the tree representing an axiom $f(a)$ is bound by a constant. Following this assumption the time of processing a single axiom (i.e. performing $D := apply(D, f(a), \wedge)$) is dependent on the size of the tree obtained till now (size of D). Use of bisection corresponds to reducing the time from the area greyed in Fig. 7a to the area greyed in Fig. 7b, as two (smaller) trees are being created, *join* operation performed (in $O(1)$ time) and finally the gluing axiom is added to the result tree (the final stripe in Fig. 7b). Iterative use of bisection provides further reduction, allowing for reaching time of execution proportional to $k \lg^2 k$ for pure taxonomies (i.e. pure-tree hierarchies of concepts) where k is the number of concepts in the terminology.

This theoretical estimation is confirmed by results of practical tests. Time of processing pure-hierarchy terminologies is showed in Table 1. The tests were performed on Pentium 4 system with 1GB of RAM

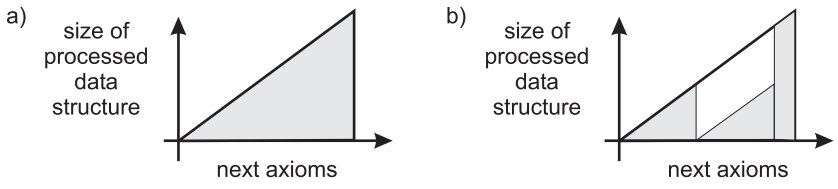


Fig. 7. Illustration of execution time: (a) without bisection, (b) with single bisection

and 3GHz clock. Ontologies of more than 5000 concepts have been generated automatically. As it can be seen, the tests confirm the quasi-linear time of creation. The progress comparing to the original algorithm (called *ACMC*) is dramatic, as ACMC execution time was proportional to k^3, which resulted in processing 500 concept terminology in about 2 days and 1000 concept terminology in over 10 days.

Table 1. Time of creation of map of concepts for taxonomies for various number of concepts

Number of concepts	Creation time [s]
3357	72
82706	392
184086	973
545450	3639

4 Applications of map of concepts

Knowledge Management Group at Gdańsk University of Technology (called KMG@GUT) [5] develops various ideas in the field of knowledge management based on the Cartographic Approach. In this section we present most interesting and advanced results.

4.1 Terminology visualization

Map of concepts can be used to present to a human user relationships among concepts. Originally developed form of map of concepts was actually a graphical form. In the course of work led by KMG@GUT the algorithm called *EnergyDots* for terminology visualization has been developed. The algorithm uses binary representation of map of concepts. It transforms the binary representation to a bidimensional picture easily readable by humans. This algorithm is based on a method of graph drawing described in [6]. Nodes of graphs are represented as "dots" (small circles) being reification of regions (atomic areas corresponding to columns in concept signatures).

The original method described in [6] uses the notion of "force field" influencing graph nodes. There are two kinds of forces influencing nodes and these are: repulsive forces (every two nodes repulse each other) and attractive forces (every two nodes connected with an edge attract each other). By simulation of force influence, the state corresponding to minimal energy potential is being gradually established. Minimal energy potential is chosen in a way fulfilling esthetical criteria.

The *EnergyDots* algorithm adapts notions of force field and repulsive and attracting forces. Repulsive forces between "dots" are computed analogically as in [6]. The difference is in the way of calculating attractive forces. During this calculation the list of concept signatures is read. For each concept a signature $s(C)$ is retrieved. "Dots" corresponding to regions with "1" in signature $s(C)$ attract each other to a common pole whose coordinates are calculated as an average of coordinates of relevant "dots".

The output of the algorithm is an arrangement of "dots" on the plane. For readability, "dots" corresponding to regions belonging to cartographic concepts can be distinguished by various colours.

Initial version of the algorithm gives good results. Exemplary result of *EnergyDots* algorithm for a simple ontology is depicted in Fig. 8. Time of executing the algorithm for a taxonomy is proportional to $n \log n$, where n is the number of regions.

4.2 Maps of concepts and inference tasks

Maps of concepts were originally implemented in KaSeA system (*Knowledge Signature Analyser*) used as a knowledge management subsystem in PIPS (*Personalised Information Platform for life and health Services*) project [7] carried out within the 6th Framework Programme of European Union, area Information Society Technologies, priority E-health. In the PIPS system, a map of concepts supports tasks of inference from terminology. Inference can be carried out in a simple way, using signatures and relationships between signatures and concepts (e.g. query about equivalence of concepts C and D can be resolved to checking if signatures are equal: $s(C) = s(D)$, other inferences being performed analogously).

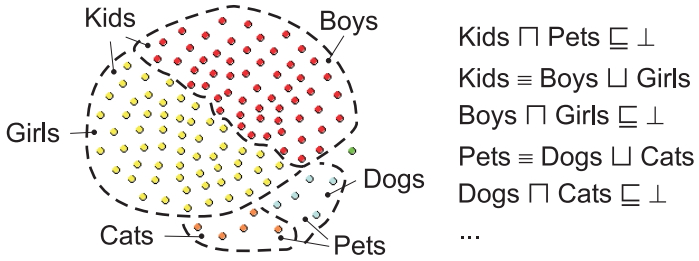


Fig. 8. Terminology visualization by using *EnergyDots* for *Pets* terminology

Very important feature of Cartographic Approach in PIPS is possibility of storing knowledge about numerous individuals and conclusions about them in an effective way. The area on the map of concepts assigned to an individual describes the knowledge about membership of the individual to the appropriate concepts. Map of concepts also allow answering queries invoking non-standard inferences in a simple way (in time proportional to signatures size and the number of different signatures). Because the whole knowledge base (individuals, concepts, roles and signatures) are stored in an Oracle database, efficiency can be additionally improved by database optimization techniques. Recent experiments are described in [8].

4.3 Ontology merging and data integration

Also, methods of describing external data sources with knowledge layer, allowing to ask queries to data sources in a way analogical as for inference system, have been developed [9]. Mappings between data and ontology are created by assigning to signatures corresponding queries understandable by a data source. The presented problem concerns fetching data from data sources on demand in terms of ontology that describes these sources.

However, integration of knowledge demands development of more advanced techniques of integration of various ontologies and queries processing in a distributed environment [10, 11]. For this issue, Cartographic Approach is used to calculate similarity of regions for different ontologies. These similarities allow to define regions for global ontology that is able to "understand" all terms in local ontologies [12]. Each region in the global ontology is additionally assigned a numerical value from the range $[0, 1]$ called a "satisfiability factor". This value reflects the knowledge of the knowledge base on satisfiability of the region. The less value of satisfiability factor, the less is the chance that there exists an individual that belongs to this region. Satisfiability factors are used in the process of responding to queries addressed to the global ontology.

4.4 Trust issues

Within the framework of PIPS project, it is of utmost importance that facts (both: terminological and assertional) come from trusted sources. However, a level of trust may be different for different knowledge sources. So, considerable amount of work concentrates on trust issues for terminologies and world descriptions. In this context, the classical DL model of knowledge must be enriched with possibility of expressing trust issues with respect to both assertions and axioms [13].

Nevertheless, building a model is not sufficient. There is also a need to develop a way of representing trust in knowledge [14, 15]. This way of representation is also developed on the basis of Cartographic Approach. In the trust-aware framework, individual and concept signatures are not binary signatures any more. A signature consisting of "0"s and "1"s is only a specific case of a general signature consisting of real numbers from the range $[0, 1]$. In Cartographic Approach, "0" at a specified position at an individual's signature is interpreted as certainty of the knowledge base that a particular individual does not belong to the region corresponding to this position; and "1" at a specified position of the signature is interpreted as a possibility (but not certainty) of the fact that the individual belongs to the specified region. In the case of such generalized, signature, the less value at a specified position, the less level of certainty (trust) of the fact that the individual belongs to the region corresponding to this position. For signatures of concepts, "0" means that a concept does not encompass the specified region, while "1" means that the concept does encompass the region. Analogically, as in the case of individual's signatures, the less value at a specified position of a signature, the less level of trust to the fact that the specified region belongs to the concept.

5 Summary

In the paper we presented *TreeFusion*, a new algorithm to create a map of concepts for DL terminology. For common taxonomy-like ontologies it allows for quasi-linear processing time in function of the number of axioms in a terminology. Implementing *TreeFusion* allowed for smooth application of KaSeA system within the framework of the PIPS project. Tests performed so far show that the algorithm allows for processing terminologies with a large number of concepts and axioms, which is of crucial importance for modern Web-based real-life applications. It is of utmost importance that the time efficiency of the algorithm seems to allow for dynamic changes in a terminology, which are not allowed in the current implementation of KaSeA due to its inherent capability of storing as much of conclusions as possible at the stage of ontology loading (for details see [1],[2]).

The algorithm exploits OBDD diagrams, originally developed for VLSI circuits design. However, new operations and transformations of OBDDs had to be invented to adapt it to ontologies processing. New method of indices ordering gave also promising results.

TreeFusion has extended an area of possible applications of Knowledge Cartography, e.g. towards DL-based ontologies integration and knowledge sources trust-awareness. The tools mentioned in this paper are being presently developed and used in the FP6 PIPS project. Moreover, they are used for education at Gdańsk University of Technology, which fosters the Semantic Web initiative among young computer engineers.

References

1. Goczyła K., Grabowska T., Waloszek W., Zawadzki M.: *The Knowledge Cartography—A new approach to reasoning over Description Logics ontologies.* In: SOFSEM 2006: Theory and Practice of Computer Science, LNCS 3831, 32nd Conference on Current Trends in Theory and Practice of Computer Science, Eds.: J. Wiederman, G. Tel, J. Pokorny, M. Bielikova, J. Stuller, 2006, pp. 293–302.
2. Goczyła K., Grabowska T., Waloszek W., Zawadzki M.: *The Cartographer Algorithm for Processing and Querying Description Logics Ontologies.* LNAI 3528: Advances in Web Intelligence, 3rd International Atlantic Web Intelligence Conference, Springer Verlag, 2005, pp. 163–169.
3. Bryant, R. E.: *Graph-based algorithms for boolean function manipulation*, IEEE Transaction on Computers, 1986.
4. Bryant, R. E.: *Symbolic Boolean Manipulation with Ordered Binary-Decision Diagrams*, ACM Computing Surveys, Vol. 24 No 3, 1992, pp. 293–318.
5. Knowledge Management Group at Gdańsk University of Technology, KMG@GUT, `http://km.pg.gda.pl/kmg`
6. Fruchterman T. M. J., Reingold E. M.: *Graph Drawing by Force-directed Placement*, In: Software—Practice and Experience, Vol. 21 No 11, 1991, pp. 1129–1164.
7. Goczyła K., Grabowska T., Waloszek W., Zawadzki M.: *Inference Mechanisms for Knowledge Management System in E-health Environment*, In: Software Engineering: Evolution and Emerging Technologies, Eds. K. Zieliński, and T. Szmuc, IOS Press, Series: "Frontiers in Artificial Intelligence and Applications", 2005, pp. 418–423.
8. Goczyła K., Waloszek A., Waloszek W., Zawadzka T., Zawadzki M.: *Ontological Queries Supporting Decision Process in KaSeA System*, In: Proceedings of 16th European-Japanese Conference on Information Modelling and Knowledge Bases. Ed. Yasushi Kiyoki, Hannu Kangassalo, Marie Duzi, Ostrava, May 2006, pp. 16–28.
9. Goczyła K., Zawadzka T., Zawadzki M.: *Managing Data from Heterogeneous Data Sources Using Knowledge Layer.* IFIP Working Conference on Software Engineering Techniques—SET 2006 (accepted for publication).
10. Calvanese D., Giacomo D., Lenzerini M.: *Ontology of integration and integration of ontologies*, Proceedings of the International Workshop on Description Logics, 2001.
11. Calvanese D., Giacomo D., Lenzerini M.: *A framework for ontology integration*, Proceedings of the 1st Semantic Web Working Symposium at the Emerging Semantic Web, pp. 201–214.
12. Goczyła K., Zawadzka T.: *Interrelationships between ontologies and their influence on ontology integration problem.* In: Bazy Danych Struktury, Algorytmy, Metody, Ed: Kozielski S., Małysiak B., Kasprowski P., Wydawnictwa Komunikacji i Łączności, Warszawa 2006, pp. 331–340.
13. Baader F. A., McGuiness D. L., Nardi D., Patel-Schneider P. F.: *The Description Logic Handbook: Theory, implementation, and applications*, Cambridge University Press, 2003.
14. Goczyła K., Zawadzki M.: *Analysis of trust issues in ontologies for different inference models.* In: Bazy Danych Struktury, Algorytmy, Metody. Ed: Kozielski S., Małysiak B., Kasprowski P., Wydawnictwa Komunikacji i Łączności, Warszawa 2006, pp. 341–350.
15. Russel S. J., Norvig P.: *Artificial Intelligence A Modern Approach*, Prentice Hall, 2003.