TOMASZ BOIŃSKI AND HENRYK KRAWCZYK
Faculty of Electronics, Telecommunications and Informatics,
Gdańsk University of Technology

DYNAMIC OWL ONTOLOGY MATCHING USING LEXICAL WORDNET-BASED MEASURES

SUMMARY

Ontologies are often used as a means of describing knowledge and the domain of operation of modern applications. A need arises for the ability to quickly match those ontologies to enable interoperability of such systems. This paper presents an extension to Noy and McGuiness' ontology construction methodology which should improve ontology interoperability and a lexicon-based algorithm for merging and aligning of such ontologies stored in the OWL language. The proposed similarity levels are presented and the proposed algorithm is described. Results of tests showing the algorithm possibilities are presented.

Keywords: OWL ontologies, merging and alignment, WordNet

1. Introduction

Ontologies are the basis for knowledge processing of many modern computer systems. By being a "formal, explicit specification of a shared conceptualisation"[13], they became a common language which is needed for proper communication [25].

The most popular language for ontologies and the *de facto* standard for describing ontologies is OWL [19]. With it comes a well-defined syntax and formal background based on Description Logic. OWL provides a common, formal way of describing knowledge, interfaces, processes etc. thus limiting the complexity of algorithms analysing such formalised structures.

Unfortunately, being portrayed as a solution to web heterogeneity, ontologies suffer from the same source of it and introduce heterogeneity themselves. Ontologies are being created by humans and are based on natural languages which, even when formalised, introduce contextuality and ambiguity of meanings of words. Each developer, as an individual, represents his or her point of view, thus introducing this ambiguity and further enhancing it by his or her own understanding of the domain. Some kind of unification of understanding of the domain of the application among all developers is needed, especially when teamwork is involved or the solution is designed for systems interoperability.

Many teams try to address this issue by developing methodologies or creating solutions that would limit ontology ambiguity. Upper ontologies like SUMO (Suggested Upper Merged Ontology) [21] or Cyc [4] aim to provide well-described, common, high level concepts that allows for the unification of domain ontologies through them. Methodologies like the one described by Noy and McGuiness [23], NeOn [27] or UPON [5], in turn, provide a way to create proper and consistent ontologies that could be reusable by a wider audience. However, those methodologies do not take interoperability into account.

In this paper a proposition for the complete process of ontology construction and integration is presented [2]. This paper proposes both a methodology for creating interoperability enabled ontologies and a lexicon-based algorithm allowing their fast and *ad hoc* integration.

2. Methodology

The proposed methodology extends the one described by Noy and McGuiness. The base method [23] is composed of 7 steps:

- Establish domain and boundaries of the ontology
- Reuse existing ontologies, if possible
- Identify all the key terms in the proposed ontology
- Define classes and their hierarchy
- Define class properties
- Define attributes of class properties
- Define individuals

Upon completion of the above steps, the ontology is verified and the cycle is repeated to eliminate any errors and inconsistency found during that evaluation. The process stops when the resulting ontology is satisfactory.

The above methodology was chosen because it is simple and provides an iterative-evolutionary approach to developing ontologies. The method was added to the tutorial of the Protégé [12, 22] editor which made it well-known and popular among a wide audience of ontology developers.

Unfortunately, processes such as ontology reuse and quality control are not described in detail and teamwork was not taken into account. The proposed methodology tries to addresses those issues by enabling explicit teamwork and adding support for ontology interoperability and exchange, thus taking into account the needs of a wider audience. It was implemented using OCS (Ontology Creation System) [2, 3].

2.1. Creating ontologies using OCS

The OCS[1] is a system developed at Gdansk University of Technology Faculty of Electronics, Telecommunications and Informatics, to provide a platform for ontology creation, integration, storage and versioning. It extends Noy and McGuiness' methodology by providing ways for distributed ontology development and its distribution and exchange through the included ontology repository.

Figure 1 presents the proposed model of developing ontologies in a distributed environment by providing a way to submit and accept changes to the ontology. The ontologies can be either public or private. Every user of the Internet can provide changes to any ontology stored within the OCS. If the ontology is private then the owner of the ontology (or a designated expert) has to accept the changes and is responsible for creating the new version.

Such a model allows for both the inclusion of changes from the community and the method for keeping the ontology consistent. Furthermore, by accepting changes from multiple sources the ontology represents a common point of view on a given subject thus increasing the possibility of its reuse and integration with other solutions.

1 http://ocs.kask.eti.pg.gda.pl

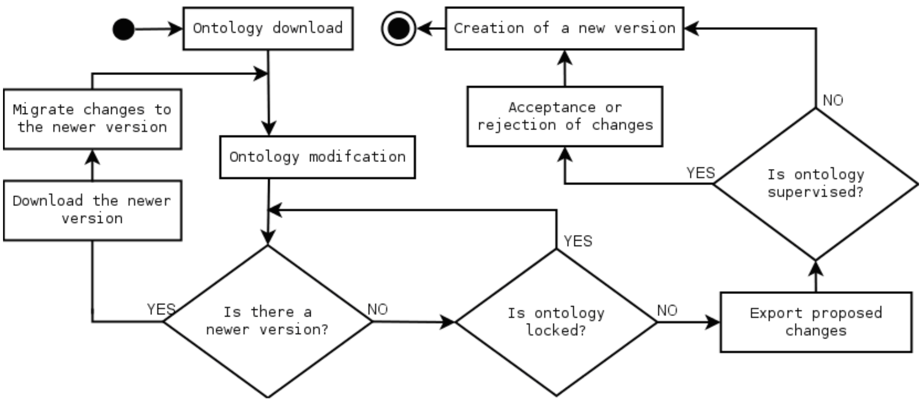

Figure 1. ***Work-flow presenting the integration of changes into an ontology***

2.2. Selection of Basic Concepts

The Noy and McGuiness methodology advises the proper identification of concepts that should be included in the ontology. In this paper, we propose to take further steps and also include into the process the analysis of core concepts in other ontologies from the same domain.

Ontologies as entities usually representing an individual point of view easily introduce heterogeneity even though a formalised language for their representation is used. The reason for that is the ambiguity of natural language and independence in creation of ontologies. Research shows that a common set of core concepts can introduce a similarity big enough for further interoperability between ontologies [2]. Building ontologies around a common set of core concepts, despite their different definitions obtained from different sources, can lead to increased probability of positive ontology integration.

Five ontologies were created using this approach[2]. Three of them, based on ENISA [6] lexicon, NIST [14] lexicon and Sommerville's "Software Engineering" book [26], were combined into one module (Risk Core Concepts module). Two others, one based on Avižienis' taxonomy [1] (Basic Security Concepts module) and one on Firesmith's taxonomy [10, 11] (Safety and Security Requirements module) were treated as separate modules. The ontologies were based on the following set of core concepts: attack, threat, protection, security, safety, safeguard, risk, asset, harm, vulnerability, threat, security feature, availability, integrity and confidentiality. All those concepts can be found in knowledge sources of all individual ontologies, albeit with different definitions.

The first three ontologies were then combined into a single ontology, which in turn was integrated with the last two, creating one, unified security and safety ontology [2]. The integration has been done manually with the help of the Falcon-AO tool [16]. Despite the different definitions, a common set of core concepts allowed for easy ontology integration as they induced high level concept similarity [2].

The integration proved that despite being based on different sources the ontologies can be combined together into one entity and provide facilities for future extension and integration with other ontologies. Furthermore, a common set of core concepts proved to be useful when working in conjunction with semantic taxonomies like WordNet [9], allowing the creation of new approaches to ontology integration and reuse.

2 Available in OCS Portal (http://ocs.kask.eti.pg.gda.pl) and at http://kask.eti.pg.gda.pl/projekty/#Ontologia_bezpieczestwa in OWL format.

3. WordNet Based Algorithm for Ontology Integration

Observations gathered during the creation of Security and Safety Ontology, as described in the previous section of this paper, led to the creation of a lexicon-based algorithm for ontology integration.

During the manual integration, knowledge represented by two ontologies was merged based on the ability to compare their elements - such as classes, individuals, relations - and properties, based on measures obtained through use of the Falcon-AO tool, were used as well as human judgment. For automation, merging a set of measures was defined based on the pragmatic approach represented by Hovy [15] and Euzenat and Volchev [8]. The proposed measures are derived from them by extending them with the possibilities introduced by modern lexicons such as WordNet.

3.1. Similarity Measures

Four complementary similarity measures are being proposed [2].

Lexical similarity P_{lex} *of classes* K_i *(i = 1, 2) and individuals* B_i *(i = 1, 2)* - basic measure of similarity between concepts. Classes and individuals' relations respectively are being derived from the WordNet structure based on direct look-ups in the dictionary. If the given concepts are found they can be marked as:

- identical when they belong to the same synset,
- disjoint when they share a common parent concept,
- logically encapsulating one another when one concept lies higher in the WordNet hierarchy.

Based on those characteristics one can determine class hierarchy and membership of individuals.

Semantic similarity P_{sem} *of classes* K_i *(i = 1, 2) and individuals* B_i *(i = 1, 2)* - secondary measure used when direct WordNet look ups are not possible. Semantic similarity based on either the Lin algorithm [18] or, when a given concept is not present in the taxonomy, the Levenhstein edit distance [17] is being used.

This similarity measure returns values from the range [0, 1] and allows one to determine whether concepts are similar, disjoint or logically encapsulating one another. In the second case, where the Levenshtein distance is used, only similarity and disjointedness of concepts can be calculated.

For resolving similarity and disjointedness, a border value of P_{sem} equal to 0.7 was established. Concept pairs for which $P_{sem} < 0.7$ are considered different and thus disjoint. All concepts where $P_{sem} \geq 0.7$ are considered equal. The border value of P_{sem} is based on a comparison of results of Lin similarity sim_{Lin} and human judgment sim_h. The comparison was done using Miller Charles' word-pairs [20] as well as a group of 23 people of varying age, nationality and occupation (Table 1). The obtained results are consistent with those gathered by other research groups, i.e. developers of Falcon-AO.

In most cases sim_{Lin} correlates with sim_h. Similarity sim_h of the pairs *tool - implement* and *brother - monk* were lower than expected (in comparison to sim_{Lin}) among people not closely related to the English language. However residents from Northern Ireland, English teachers etc. marked those pairs as highly similar.

In case of the pairs *bird - cock* and *bird - crane*, where the second lemma is logically encapsulated by the first lemma, sim_{Lin}, 0.74 and 0.72 respectively, is very close to the border value. Human testers decided, however, that having only the value of similarity between those words, they can be considered as equal.

Table 1. ***Comparison between Lin similarity (sim_{Lin}) and human judgment (sim_h) on Miller and Charles' [20] test-pairs. $ident_{\%}$ describes the percentage of human subjects that found the two given words equal.***

$Lemma_1$	*$Lemma_2$*	*$ident_{\%}$*	*sim_h*	*sim_{Lin}*
car	automobile	100%	0.92	1.00
gem	jewel	82.61%	0.78	1.00
journey	voyage	100%	0,90	0.80
boy	lad	82.61%	0.77	0.92
coast	shore	95.65%	0.85	0.97
asylum	madhouse	69.57%	0.68	0.86
magi-cian	wizard	82.61%	0.84	1.00
midday	noon	82.61%	0.81	1.00
furnace	stove	69.57%	0.73	0.25
food	fruit	30.43%	0.46	0.18
bird	cock	34.78%	0.40	0.74
bird	crane	26.09%	0.37	0.72
tool	implement	56.52%	0.54	0.95
brother	monk	34.78%	0.42	0.97
crane	implement	8.70%	0.26	0.38
lad	brother	26.09%	0.29	0.25
journey	car	4.35%	0.16	0.00
monk	oracle	4.35%	0.22	0.22
cemetery	woodland	0.00%	0.09	0.13
food	rooster	17.39%	0.25	0.09
coast	hill	0.00%	0.16	0.64
forest	graveyard	0.00%	0.09	0.13
shore	woodland	0.00%	0.12	0.14

monk	slave	4.35%	0.09	0.24
coast	forest	0.00%	0.11	0.14
lad	wizard	0.00%	0.07	0.25
chord	smile	4.35%	0.07	0.28
glass	magician	0.00%	0.07	0.21
noon	string	0.00%	0.02	0.07
rooster	voyage	0.00%	0.02	0.00

Similarity of comments P_{kom} attached to classes K_i (i = 1, 2) and individuals B_i (i = 1, 2) - third level of similarity between ontology elements. The words belonging to comments are being mapped to nodes of a bipartite graph *G (V, E)*, where *V* is a set of nodes and *E* is a set of edges. Set *V* can be divided into two independent sets, *L* and *R*, such that each edge $e \in E$ connects node $l \in L$ with node $r \in R$. Edges of the graph are weighted with similarity between individual words calculated as in two previous cases. The value of the P_{kom} similarity is calculated as a maximal assignment in the graph divided by the number of elements in the longer of the two comments (Equation 1), which ensures that the value of P_{kom} stays within [0,1] range.

$$P_{kom} = \frac{\sum_i P_{lr_i}}{\max(|L|, |R|)} \tag{1}$$

Structural similarity P_{str} of classes K_i (i = 1, 2) and individuals B_i (i = 1, 2) - fourth level of similarity taking into account the structural alignment of a given concept with its closest neighbors, used where no other possibility can be applied. The similarity can be calculated using Equation 2, in which both the type and direction of the relation are taken into consideration.

$$P_{str} = \frac{\sum_i \min(r_i)}{\sum_i \max(r_i)} \tag{2}$$

where:
r_i - number of occurrences of relation *i* (where *i* = {subsumption, membership, equality, disjointness, union, intersection}) in which the given concept takes part.

With constant development of many lexicons, this approach seems to be increasingly justified. In the current 3.0 version of WordNet, 155,287 different nouns in the English language with 206,941 word-meaning pairs can be found [24]. It is highly probable that most of the words used in the concepts' description will be found within that lexicon. This way, connections between concepts can be derived from WordNet, thus enriching the output ontology. Further levels of similarity allow for the mapping of concepts into one another, even when they are not found in the used lexicon, making the proposed algorithm usable in general scenarios.

The aforementioned similarity measures can be combined to form a complete WordNet-based solution for concept comparison. With the constant development of WordNet-like semantic dictionaries, their effectiveness will grow. Today, with over 15,500 different nouns WordNet, with proposed similarity measures, can be used as a base for lexicon-based algorithm for ontology integration which is presented in next section of this paper.

3.2. The algorithm

The proposed algorithm [2] is based on lexical analysis of ontology components and is designed to operate on OWL ontologies[3]. The assumption was that the integrated ontologies are representing the same domain. Furthermore, two observations have been made - that in most cases ontology elements' names are usually represented by nouns, and that most of the information is either explicitly represented by classes and relations between them or can be easily derived and transformed into such a representation [2].

The algorithm takes two OWL ontologies as its input and generates one ontology as its output. The resulting ontology can be either a merge (in terms of unification of URI's in the OWL language) or an alignment (in terms of importing source ontologies and including only mapping between their elements).

The algorithm starts with comparing every element located at the same level of hierarchy in both ontologies with each other and the works recursively. A comparison function determines whether concepts are similar, disjoint or logically encapsulating based on similarity measures introduced earlier in this paper. First, lexical similarity is checked, then the semantic one and, whenever possible, the algorithm tries to determine parent-child relationship between the compared concepts. If the first two approaches will not provide a satisfactory answer about the concept's similarity then the similarity is decided based on Equation 3.

$$P_{sk} = w_1 P_{str} + w_2 P_{kom} \tag{3}$$

where:
P_{str} - structural similarity (Equation 2),
P_{kom} - semantic similarity of comments (Equation 1),
w_i - weights of both similarities. Based on experiments they were set to be as follows: $w_1 = 0.3$; $w_2 = 0.7$.

Similarly, as in individual measures, concepts with $P_{sk} \geq 0.7$ are treated as identical and concepts with $P_{sk} < 0.7$ are treated as different.

Results of the comparison govern the action performed by the algorithm. If the concepts are determined to be equal they are merged together by connecting with equivalent property and added to the output ontology. Next, their subconcepts combined together into one tree and attached to these new nodes. If the concepts are determined to be different, then they are added with their subtrees to the output ontology. No further analysis of those subtrees is performed. If one of the concepts is determined to be more general than the other, it is added to the resulting ontology and the more specific one is placed in its subtree.

4. Results

The proposed measures and the algorithm were tested using selected ontologies developed by the Ontology Alignment Evaluation Initiative as input for EON Ontology Alignment Contest [7] and security ontology [2], also described briefly in Section 2.2.

4.1. OAEI ontologies

3 Implementation available through OCS system (http://ocs.kask.eti.pg.gda.pl).

The test ontologies are describing the Bibtex structure[4]. They were compared with a reference one. In one case, an unrelated ontology describing food and wine was used. The following scenarios were considered:

- Merging with an identical ontology - all concepts were merged with final similarity equal to 1.0. The algorithm introduced one additional connection (between *Address* and *Reference*) due to the domain overlap, with source ontologies not providing any additional information about the concepts except their labels;
- Merging with completely different ontology - the source ontology was merged with a completely irrelevant one (describing food and wine) and all concepts in both ontologies were correctly determined to be different;
- Merging with similar ontologies stored using more general dialects of the OWL language - all concepts within the original and generalised ontologies were marked as identical with a similarity equal to 1.0. Identically, as in the first test, one additional connection was introduced;
- Merging with identical ontology with removed labels (comment and structural similarity only) - the ontology was merged with identical but with the labels replaced by random, meaningless strings. The algorithm based its work solely on the structure of both ontologies and comments attached to their elements. Most of the connections were determined correctly. The algorithm, however, introduced two additional connections between concepts *InCollection* and *Chapter* and their respective matches in the scrambled ontology (*dcsqdcsqd* and *dzqndbzq*). Both of these concepts were surrounded with identical structures and had similar comments attached ("A part of a book having its own title" and "A chapter (or section or whatever) of a book having its own title", respectively). Thus, the distinction between those concepts could not be determined correctly.

In all the above cases, the algorithm produced satisfactory results proving its usefulness for small, domain-oriented ontologies. Such ontologies are usually used by systems such as computer agents or web services. This way it is possible to integrate applications in which the behavior of implemented system is described by such ontologies enabling their interoperability without the need for the manual unification of the descriptions of those systems.

4.2. Security and Safety Ontology

The security ontology was at first created manually. Later, the integration process was repeated using the proposed algorithm. First, the three smaller ontologies (ENISA, NIST and the one based on Sommerville’s book) were combined into a single module, and afterward the three modules were combined into one ontology. The obtained ontologies were compared with results of manual integration.

During creation of the Risk Core Concepts module 1,170 comparisons between 153 classes were performed, out of which 13 were incorrect (1.11%). Some of the incorrect matches resulted from errors in source ontologies. Those errors were corrected. During integration of the three modules into a single ontology 1,956 additional comparisons were performed. Of those, 31 were incorrect (1.59%), and the resulting ontology was very similar to the one obtained manually. Additionally, usage of the algorithm allowed for the discovery of some errors and oversights done during manual integration and helped to improve the resulting ontology.

Large ontologies often cross the boundary of being domain-oriented, thus the possibility of automatic lexical-based integration is limited. Still, in cases where domains of integrated ontologies are similar lexical-based integration proves to be useful by limiting the amount of manual connections that need to be performed and allows for the verification of both input ontologies and the

4 Available at http://oaei.ontologymatching.org/2006/benchmarks/

result of integration.

5. Conclusions and Future Works

The proposed algorithm uses WordNet as the underlying lexicon for concept comparison. The tests performed have proved its usability in performing the integration of both small and large domain ontologies, especially when backed by the proposed methodology for ontology construction.

The proposed methodology for ontology development helps the unification of understanding of the domain which is being covered by the ontology. Common agreement upon used vocabulary and definitions is crucial for knowledge exchange and teamwork using Internet applications. Proposed similarity measures and the algorithms offer a means of providing such an agreement, even when different knowledge sources and concept definitions are used, thus increasing possibilities of interoperability between current and future installment of ontology-based systems.

In the future it is possible to replace WordNet with another lexicon when another emerges, especially since references to WordNet's database are considered slow. Further extension of WordNet (or similar) taxonomy can increase the quality of the algorithm and loosen the strict bonds of assumption of domain orientation of the integrated ontologies. Further tests should also be performed to tweak the boundary value of treating concepts as different or equal and extend the proposed similarity values to encapsulate the new findings and observations.

Acknowledgements

This work was funded by National Science Center under grant No. N516 476440.

6. Literature

[1] Avizienis, A., Laprie, J.C., Randell, B., and Landwehr, C. *Basic concepts and taxonomy of dependable and secure computing*. Dependable and Secure Computing, IEEE Transactions on, 1(1):11–33, 2004.
[2] Boiński, T.: *Procedures for merging and alignment of domain ontologies* [in Polish], PhD thesis, Faculty of Electronics, Telecommunications and Informatics, Gdansk University of Technology, 2012.
[3] Boiński, T., Budnik, Ł., Jakowski, A., Mroziński, J. and Mazurkiewicz, K.: *OCS - domain oriented ontology creation system*, Polish Journal of Environmental Studies, Vol. 18, No. 3B, pp. 35-38, 2009.
[4] Cycorp Inc. (2010), *Cyc* [Online]. Available at: http://www.cyc.com/cyc
[5] De Nicola, A., Missikoff, M. and Navigli, R.: A software engineering approach to ontology building, Information Systems, Vol. 34, No. 2, pp. 258-275, 2009.
[6] ENISA (2010), *Enisa: a European Union Agency - Glossary of Risk Management* [Online]. Available at: http://www.enisa.europa.eu/act/rm/cr/risk-management-inventory/glossary
[7] Euzenat, J. (2006): *EON Ontology Alignment Contest* [Online]. Available at: http://oaei.ontologymatching.org/2006/
[8] Euzenat, J. and Valtchev, P.: *Similarity-based ontology alignment in OWL-lite*, ECAI 2004: 16th European Conference on Artificial Intelligence, August 22-27, 2004, Valencia, Spain: including Prestigious Applicants of Intelligent Systems (PAIS 2004): proceedings, 2004.
[9] Fellbaum, C.: *WordNet: An electronic lexical database*, The MIT press, 1998
[10] Firesmith, D. G.: *A Taxonomy of safety-related requirements*, International Workshop on High Assurance Systems (RHAS'05), 2005.
[11] Firesmith, D.G.: *A taxonomy of security-related requirements*, International Workshop on High Assurance Systems (RHAS'05), 2005.

[12] Gennari, J. H., Musen, M. A., Fergerson, R.W., Grosso, W. E., Crubézy, M., Eriksson, H., Noy, N. F. and Tu, S.W.: *The evolution of Protégé: an environment for knowledge-based systems development*, International Journal of Human-Computer Studies Vol. 58, No. 1, pp. 89-123, 2003.
[13] Gruber, T. R. et al. *A translation approach to portable ontology specifications*, Knowledge acquisition, Vol. 5, p. 199, 1993.
[14] Guttman, B. and Roback, E. A.: *An introduction to computer security: the NIST handbook*, DIANE Publishing, 1995.
[15] Hovy, E.: *Combining and standardizing large-scale, practical ontologies for machine translation and other uses*, Proceedings of the 1st International Conference on Language Resources and Evaluation (LREC), pp. 535-542, 1998.
[16] Jian, N., Hu, W., Cheng, G. and Qu, Y.: *Falcon-AO: Aligning ontologies with Falcon*, Integrating Ontologies Workshop Proceedings, 2005.
[17] Levenshtein, V. I.: *Binary codes capable of correcting deletions, insertions, and reversals*, Soviet Physics Doklady, Vol. 10, pp. 707-710, 1966.
[18] Lin, D.: *An information-theoretic definition of similarity*, Proceedings of the 15th International Conference on Machine Learning, Vol. 1, pp. 296-304, 1998.
[19] McGuinness, D. L., Van Harmelen, F. et al. *OWL web ontology language overview*, W3C recommendation, Vol. 10, pp. 2004-03, 2004.
[20] Miller, G. A. and Charles, W. G.: *Contextual correlates of semantic similarity*, Language and cognitive processes, Vol. 6, No 1, pages 1-28, 1991
[21] Niles, I. and Pease, A.: *Towards a standard upper ontology*, Proceedings of the international conference on Formal Ontology in Information Systems, pp. 2-9, 2001.
[22] Noy, N. F., Fergerson, R. and Musen, M.: *The knowledge model of Protege-2000: Combining interoperability and flexibility*, Knowledge Engineering and Knowledge Management Methods, Models, and Tools, pp. 69-82, 2000.
[23] Noy, N. F., McGuinness, D. L. et al. *Ontology development 101: A guide to creating your first ontology*, 2001.
[24] Princeton University (2010), *WordNet Database Statistics* [Online]. Available at: http://wordnet.princeton.edu/wordnet/man/wnstats.7WN.html
[25] Russel, S. and Norvig, P.: *Artificial Intelligence: A Modern Approach*. Prentice hall. 2003
[26] Sommerville, I.: *Software Engineering*. 8th, Harlow, UK: Addison-Wesley, 2006
[27] Suárez-Figueroa, M. C. et al (2009): *D5. 4.2: Revision and extension of the neon methodology for building contextualized ontology networks, NeOn project* [Online]. Available at: http://www. neon-project. org

ŁĄCZENIE ONTOLOGII OWL Z WYKORZYSTANIEM MIAR BAZUJĄCYCH NA WORDNET

STRESZCZENIE

Ontologie coraz częściej służą jako mechanizm opisu zarówno zgormadzonej wiedzy, jak i samej domeny, na której operuje aplikacja. Ze względu na mnogość dostępnych rozwiązań zachodzi konieczność umożliwienia dynamicznego łączenia ta-

kich ontologii, co umożliwi współpracę korzystających z nich aplikacji. Publikacja prezentuje rozszerzenie metodologii wytwarzania ontologi opracowanej przez Noya i McGuinessa, które zwiększa szansę na współpracę pomiędzy systemami wykorzystującymi ontologię. Zaprezentowano również bazujący na leksykalnej analizie algorytm łączenia i odwzorowywania ontologii w języku OWL. Zaproponowane miary podobieństwa pomiędzy elementami ontologii zostały opisane w pracy. Zaprezentowano również wyniki eksperymentów potwierdzające możliwości algorytmu.
Keywords: ontologie OWL, łączenie i odwzorowywanie, WordNet

Tomasz Boiński, Piotr Orłowski and Henryk Krawczyk
Politechnika Gdańska
Wydział Elektroniki, Telekomunikacji i Informatyki
Katedra Architektury Systemów Komputerowych
ul. Gabriela Narutowicza 11/12, 80-233 Gdańsk
tobo@eti.pg.gda.pl