Concept Mapping: Connecting Educators
Proc. of the Third Int. Conference on Concept Mapping
Tallinn, Estonia & Helsinki, Finland 2008

CONCEPT MAP GENERATION FROM OWL ONTOLOGIES

Vita Graudina, Janis Grundspenkis
Department of Systems Theory and Design, Riga Technical University, Latvia
{vita.graudina, janis.grundspenkis}@cs.rtu.lv

Abstract. The paper is dedicated to concept map generation from OWL ontologies. This is an important issue because creation of concept maps for knowledge assessment tools are time and effort consuming. During last years since ontology description language OWL is developed amount of created ontologies have been extremely increased. A lot of ontologies written in OWL are freely available on Internet and could be reused as a base for concept maps. Therefore these resources may be effectively used if corresponding algorithms and tools are available. The paper presents algorithms for concept map generation from OWL ontology. The proposed algorithms are implemented into a software tool which is tested for concept map generation.

1 Introduction

Concept mapping has become quite popular and is applied to different areas, for example, knowledge management (Coffey, 1999), system analysis (Freeman, 2004), and as a research tool in educational science (Markham et al., 1994). Concept maps as a specific kind of mental models play a significant role in educational settings. Concept maps have proven their value for single and collaborative learning (Novak & Gowin, 1984; Cicognani, 2000), and for student knowledge assessment and self-assessment (Ruiz-Primo, 2004; Anohina et al., 2006). Due to the focus of this paper let's highlight only the usage of concept maps for knowledge assessment and self-assessment. The concept mapping approach offers a reasonable balance between requirements to assess a learner's knowledge at higher then fourth level of Bloom's taxonomy (Bloom, 1956) and implementation complexity of computer-based assessment system. Concept maps are used for representation and measuring of individual's knowledge by visualization of a graph, i.e., concept maps are graphs which nodes and arcs represent concepts and relationships between them, respectively (Croasdell et al., 2003).

Usually concept maps are represented as hierarchies with most general concepts at the top, and more specific concepts at lower levels (Novak & Canas, 2006). Concept maps can have different topologies, such as linear, circular, hub/spoke, tree and network/net (Yin et al., 2005). A wide variety of concept maps is an additional reason why knowledge assessment tools are based on concept maps (Ruiz-Primo, 2004; Anohina et al., 2007), because they offer a wide variety of different tasks. Concept map building tasks range from high-directed to low-directed depending on information provided for students. All tasks may be divided into fill-in tasks (high-directed) where concept map's structure is given and construction tasks (low-directed) where students themselves must create a concept map's structure (Ruiz-Primo, 2004).

Concept map based tasks as test items for assessment allows seeing student's cognitive structure, i.e., their knowledge structure. Thus, concept mapping promote system thinking which frequently is a critical point for students. Yet more, the use of concept maps supports a process oriented learning, in which a teacher divides a study course into several stages, i.e., logically complete parts, for example, topics (Anohina et al., 2006). Thus, a systematic assessment is supported, which, in its turn, allows to change teaching strategies and the learning content timely depending on results of assessment.

Concept maps are quite similar with ontologies which usage in computer science during last two decades has been rapidly increased. The most important step in ontology evolution is the development of Web Ontology Language (OWL) in the year 2004 (Bechhofer et al., 2004). OWL development together with many tools for ontology construction (Protégé, WebODE, OntoStudio etc.) made ontologies quite widespread and the number of available ontologies is fastly growing. On the Internet there are more than 83 000 ontologies encoded in OWL (searched using www.google.com in June, 2008) comparing with 58 000 in April, 2008 and 33 000 in September, 2007.

Ontologies are used in computer based tutoring systems for several purposes. Representation of a particular subject (Vergara Ede & Capuano, 2003; Bakhtyari, 2006), curriculum (Doan & Bourda, 2006) and student model (Aroyo & Dicheva, 2002) are only some examples. Such intelligent tutoring systems as FLUTE (Devedzic et al., 2000) and SlideTutor (Crowley et al., 2003) use several ontologies mainly in pedagogical and expert modules. The use of ontologies in education follows several goals. First, an ontology like a concept map represents a knowledge structure. Second, an ontology may support reasoning for diagnosis of causes of learner's mistakes and misconceptions, which is a relevant functions of student diagnosis module. Moreover, each item of the ontology may be supplied with references to corresponding learning objects which may be used by students to correct their mistakes. Third, an ontology can represent not only definite concepts and semantics of their relationships but also all synonyms of both. This may rise flexibility and adaptability of knowledge assessment allowing students to use synonyms. And last, but not least, at the moment on the Internet there are available quite a lot ontologies that correspond to taught subjects. Their usage may help teachers who are creating courses to reach compatibility of the knowledge structure they wanted to create with corresponding ontology. Our experience confirms that for teachers it is much easier to edit a concept map generated from an ontology or even to build it from scratch instead of mastering formal ontology languages and specific tools for ontology construction. Especially this is important for teachers who already use knowledge assessment tool based on concept maps (Anohina & Grundspenkis, 2007; Anohina et al., 2007).

Survey of literature showed that the basic technology for ontology transformation to other formats is based on metamodels. There are already defined metamodels for mapping from UML (Unified Modelling Language) to OWL, from OWL to UML, from OWL to Topic Maps, from Topics Maps to OWL, from OWL to Common Logic within Meta-Object Facility Query/View/Transformation framework (OMG, 2006; Na et al., 2006). Algorithms for ontology transformation into schema of relational database also are worked out (Vysniauskas & Nemuraite, 2006). Transformation from ontology to concept map and vice versa is implemented in COE tool (http://cmap.ihmc.us/coe) for ontology visualization, editing and building using concept maps. However, this tool is not suitable for the developed knowledge assessment system (Anohina & Grundspenkis, 2007; Anohina et al., 2007) due to following reasons. First, its user interface and automatically added names of links and nodes (such as "is a", "are", "(DataTypeProperty)", "Things which", etc.) don't support Latvian language. Second, the full functionality of COE, for example, various restrictions on values, cardinalities, etc. is not necessary for concept maps used for knowledge assessment. Third, the transformation tool must be compatible with already developed knowledge assessment system.

The remaining of the paper is organized as follows: in Section 2 similarities between OWL ontologies and concept maps are outlined. Proposed algorithms for concept map generation from OWL ontology is described in Section 3. In Section 4 testing results of concept map generation tool in which proposed algorithms have been implemented are given. Finally, some conclusions and future work is outlined.

2 Similarities between OWL ontologies and concept maps

Both ontologies and concept maps represent some domain. Both have classes or concepts and relations between them. Unlike concept maps ontologies have also attributes for classes, their values and restrictions on them. In other words ontologies are more expressive. At the same time concept maps also could represent the same more expressive features using only concepts and links between them. Since ontologies are described using special languages, such as OWL (Bechhofer et al., 2004), DAML+OIL (van Harmelen et al., 2001), Ontolingua (Gruber, 1992), or special knowledge structure like frames correspondence between ontology elements and concept map elements should be defined.

Mappings between main elements of OWL ontology and elements of the concept map are shown in Figure 1. Main elements of OWL ontology which correspond to the concept are a class, an instance, a datatype property, its value and a type of value, while an object property corresponds to a link. An object property is semantic relation between classes or instances.

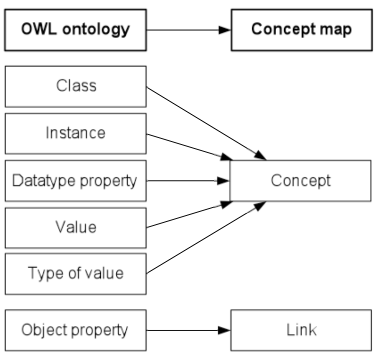


Figure 1. Correspondence of main ontology elements to concept map elements

However, OWL ontology has also other constructions to define elements which could be transformed as concept map elements. OWL constructions and corresponding concept map elements are summarized in Table 1.

#	OWL elements	Concept map elements	#	OWL elements	Concept map elements
1.	`owl:Class`	Concept	7.	`owl:FunctionalProperty`	Link
2.	`owl:DatatypeProperty`	Concept	8.	`owl:InverseFunctionalProperty`	Link
3	`owl:DeprecatedClass`	Concept	9.	`owl:ObjectProperty`	Link
4.	`owl:DeprecatedProperty`	Link	10.	`owl:SymmetricProperty`	Link
5.	`owl:equivalentClass`	Class	11.	`owl:TransitiveProperty`	Link
6.	`owl:equivalentProperty`	Link			

Table 1. Correspondence of specific OWL elements to concept map elements

Along with links retrieved from object properties additional links are introduced:

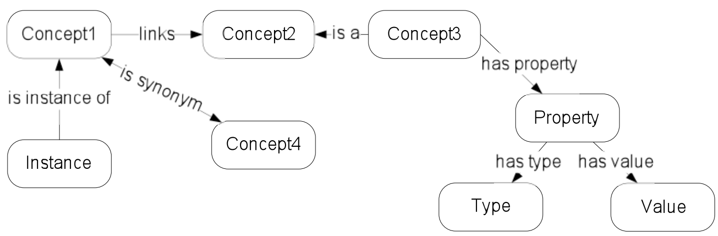
- "is a" to represent hierarchal relation between two classes;
- "is instance of" to represent relation between a class and its instances;
- "is synonym" to represent that two classes/instances are synonyms;
- "is not" to represent that two classes are complement;
- "has property" to present relation between a class/instance and a datatype property;
- "has type" to present relation between a datatype property and a data type of its value;
- "has value" to present relation between a datatype property and its value.

Mentioned additional links are needed to represent relations between concepts in the concept map which are described using specific constructions in the ontology. Detailed information about these constructions can be found in (Graudina, 2008).

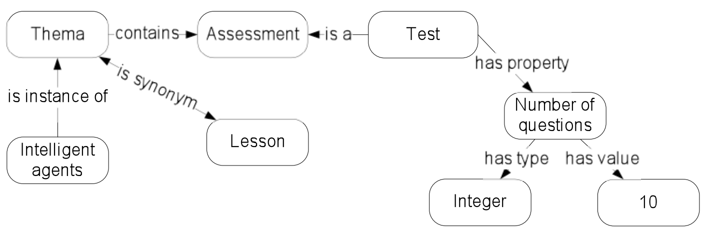
3 Algorithm for concept map generation from OWL ontology

In this section the concept map generation algorithm is described. There is a need for text analyzer (a parser) to mine information from an ontology described in OWL, which is saved as text in *.owl. The parser mines concept names and their relations analyzing *.owl file. The algorithm used by the parser is described in this section.

Information mined from ontology is stored in an incidence matrix, where names of concepts and relations between concepts (the names of the links and direction of the links) are stored. An example of an abstract concept map is shown in Figure 2a). The concrete example for the abstract concept map is given in Figure 2b). The corresponding incidence matrix for the abstract concept map is given in Table 2.



a)



b)

Figure 2. Example of concept maps a) abstract, b) concrete

Name of concept	Concept1	Concept2	Concept3	Concept4	Instance	Property	Type	Value
Concept1		link		is synonym				
Concept2								
Concept3		is a				has property		
Concept4	is synonym							
Instance	is instance of							
Property							has type	has value
Type								
Value								

Table 2. The incidence matrix

Basic steps for concept map generation from an ontology are the following:

1. Read an ontology file and check OWL syntax.
2. Find all classes (begin creating of incidence matrix).
3. Find subclasses of each class (for particular class add link “is a” which goes from subclass to superclass in the matrix).
4. For each class check equivalence and similarity (add link “is synonym” in the matrix).
5. For each class check intersection and union to other classes (add link “is a” in the matrix).
6. For each class check complement relations to other classes (add link “is not” in the matrix).
7. Find instances of each class (add instances and links “is instance of” between appropriate classes and instances which go from a instance to a class to the matrix).
8. Find datatype properties for each class and instance (add properties and links “has property” between appropriate class/instance and a property to the matrix).
9. Find values for each datatype property (add properties’ values and links between a datatype property and its value “has value” to the matrix).
10. Find types for datatype property’s values (add a value’s type and a link between a datatype property’s value and its type to the matrix).
11. Find object properties for each class/instance (add appropriate links between classes or instances to the matrix).
12. Check if an object property is symmetric or transitive (extend the matrix with appropriate links).
13. Perform correction of concept and link names (replace understrike sign “_” with space).
14. Display completed incidence matrix as a graph.

At first, the teacher chooses an ontology for concept map generation, then the concept map generation software performs all transformations and displays a generated concept map. Before transformation actions verification of chosen *.owl file is performed to check correctness of used OWL syntax. If the ontology doesn’t conform OWL syntax, the teacher receives an error message. If verification is successful the concept map generation algorithm starts. In the beginning all classes, their hierarchy and instances are found, afterwards datatype properties, their values and types are found, and at the end object properties are found. Finally, correction of concept and link names is performed and generated concept map is displayed to the teacher.

For simpler implementation previously mentioned algorithm’s steps are merged in the following way: transformation of ontology classes and instances (steps 2, 3, 4, 5, 6, 7), transformation of datatype properties, their values and types (steps 8, 9, 10), transformation of object properties (steps 11, 12).

The first step of the concept map generation algorithm is to find a class hierarchy and its extension with instances of classes (see Figure 3). At first the root class is found, and a class name is added to the incidence matrix. Then it is checked if the class has instances. If yes, then names of instances are added to the matrix as well as the link “is instance of” to relate instances with the root class. If the root class hasn’t instances, it is checked if it has subclasses. If yes, then the names of subclasses are added to the matrix as well as the link “is a” to relate the root class with subclasses. Each subclass is checked if it has instances and subclasses similarly as in case with the root class. Based on the results of checking the matrix is updated. Instances of the class are mined before subclasses due to simpler recursion programming needed for mining. After all classes have been found they are checked to mine additional information. Classes are checked if they have synonym classes. If yes, then the matrix is updated with the link “is synonym” between synonym classes. Then classes are checked if they are result of intersection or union of other classes, if yes, then in case of intersection the link “is a” is added from the class to other classes from intersection, in case of union the link “is a” is added from classes from union to the class.

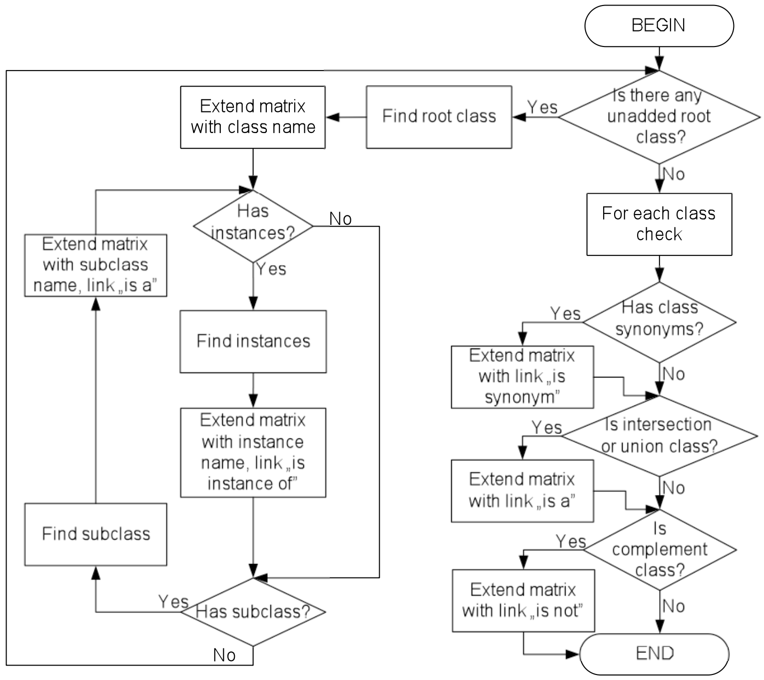


Figure 3. Transformation of ontology classes and instances

The next step is to find datatype properties, their values and types of values (see Figure 4). First, datatype properties are found for each class. The matrix is updated with names of datatype properties, as well as the link "has property" at the appropriate class. Then for each datatype property it is checked if allowed values for this property are defined. If there are such values, their names are added to the matrix, as well as the link "has value" at the datatype property. If the type of values is defined, it is also added to the matrix and the link "has type" at the appropriate datatype property. For each class which has datatype property it is checked are some of defined allowed values assigned to its instances. In case if any of values are assigned to the instance the link between the class and this value is deleted and added between the instance and the value, i.e., the value becomes local, it is related only to the instance.

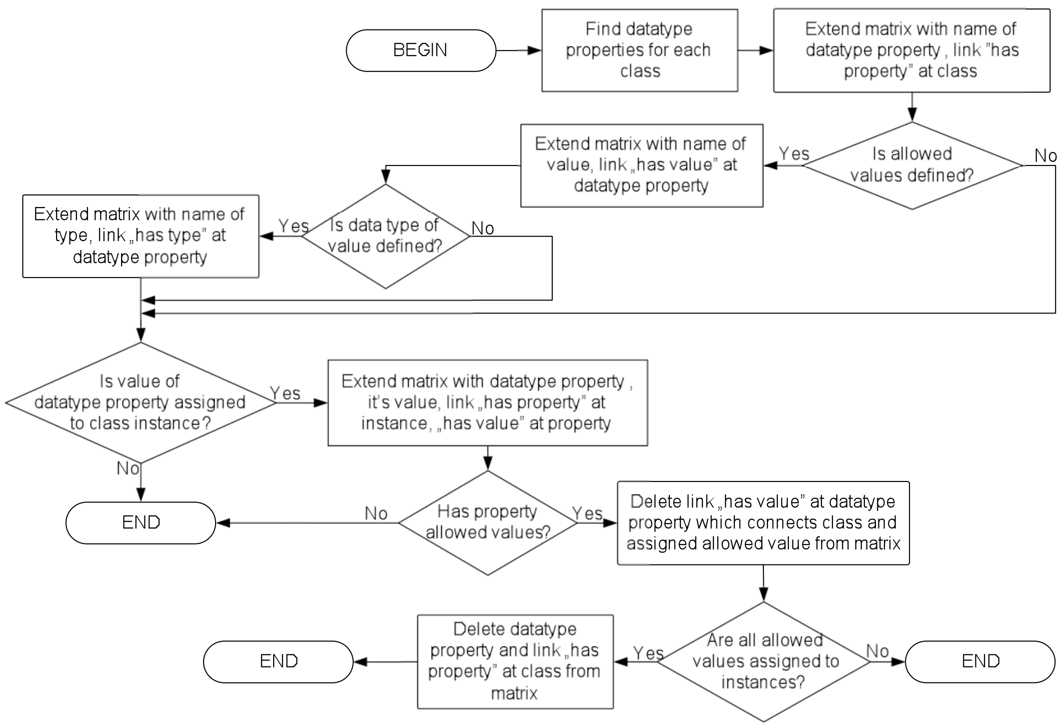


Figure 4. Transformation of datatype properties, their values and types of values

The next step after datatype property mining is object property mining. In this step semantic relations between classes and their instances are found. First, all object properties between classes are found, then domain and range classes for object properties are found and links (names of object properties) at domain classes are added to the matrix. If the object property is symmetric then the link is added also at the range class. Then it is checked if the object property is transitive. If yes, then the link is added at the “transitive” class, for example, if the class A is linked to the class B and the class B is linked to the class C, then in the matrix also the link between A and C is added, and the class A is the “transitive” class. Checking of object property’s symmetry and transitivity provides the concept map with additional links, that is, in case of the symmetric link the link in the concept map is bidirectional and in case of transitive link the third inferred link is added to the concept map. Second, object properties between instances are found. Then the matrix is updated with links between instances. However, links between superclasses of instances are deleted. Characteristic of symmetry and transitivity is checked, too. The realization of this step is shown in Figure 5.

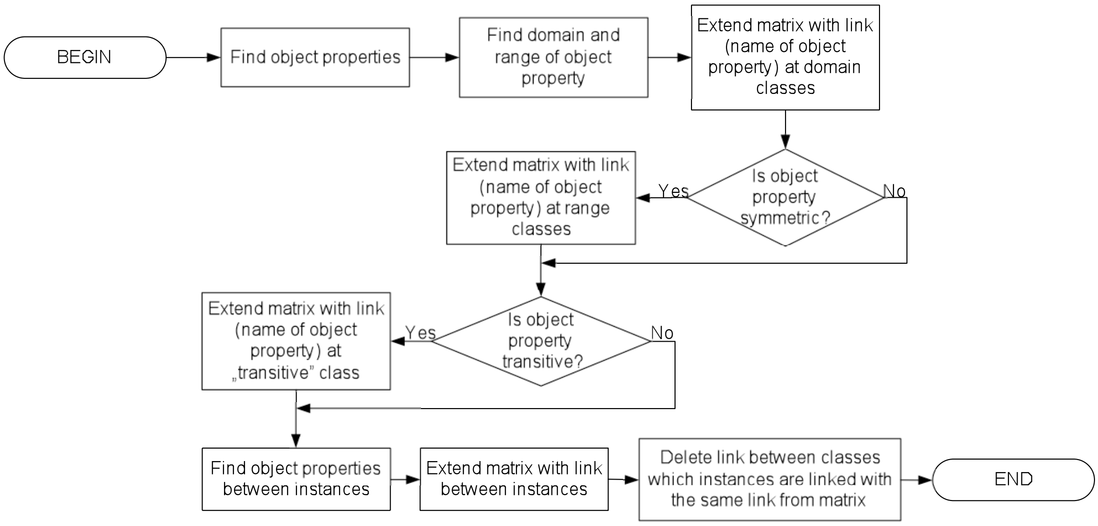


Figure 5. Transformation of object properties

Before the matrix is displayed as concept map correction of concept and link names are performed, i.e., the understrike sign “_” is replaced with space. Correction is needed because OWL doesn’t support space sign in the element names, and the world practise is to replace space sign with understrike sign “_” in OWL code, and for concept map visuality it is replaced back.

4 Implementation and testing

Proposed algorithms for concept map generation have been implemented in software. Concept map generation tool is implemented using programming language Java and application programming interface Jena (available at: http://jena.sourceforge.net/) for accessing *.owl files and parsing OWL elements. For class hierarchy and instance mining reasoning mechanisms build in Jena are used.

The developed tool was tested using 10 ontologies of the computer science field. It was unable to generate concept maps from two ontologies due to errors in OWL syntax. Two concept maps were generated with unnecessary concepts. The authors of these ontologies have used a special mode of the ontology development tool called Protégé, i.e., frame mode (available at: http://protege.stanford.edu). Series of special concepts used only in this mode are added to created ontologies. These unnecessary concepts are such as DIRECTED-BINARY-RELATIONS, PAL-RANGE, PAL-NAME, PAL-DESCRIPTION, PAL-STATEMENT, etc. which are needed to support inner structure of frames used in Protégé. So, six concept maps are successfully generated with the developed tool. One example of concept map generated using the developed tool is shown in Figure 6.

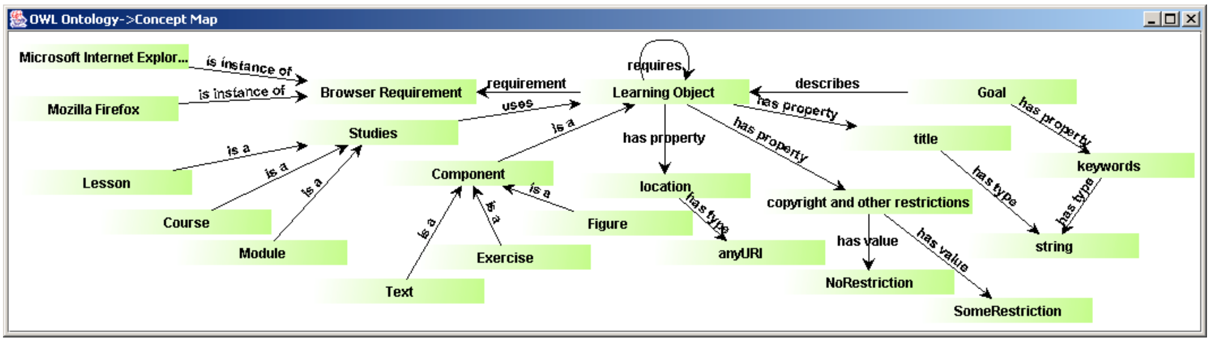


Figure 6. Generated concept map

5 Conclusions and future work

Concept maps and ontologies both represent knowledge structures and may be used to promote system thinking. Concept maps have proved their usefulness in teaching and learning process, in particular, for knowledge assessment and self-assessment. Concept maps promote process oriented learning and systematic knowledge assessment. Ontologies may increase flexibility and adaptiveness of knowledge assessment systems to each individual learner. Moreover, ontologies made a concept map construction tasks easier for teachers, and may help them to discover unrelated parts in a concept system of their courses.

In this paper algorithm for concept map generation from OWL ontologies is presented. The proposed algorithm is based on defined similarities between elements of OWL ontology and elements of the corresponding concept map. The concept map generation tool has been implemented using programming language Java and application programming interface Jena. Testing results of the tool confirmed its suitability for concept map generation.

Future work is related to further refinement and enhancement of concept map generation software. First task is to eliminate drawback connected with ontologies made in frame mode of ontology construction tool Protégé. It is valuable for concept map generation tool that all ontologies described in OWL could be transformed into concept maps independently from their construction tool. Then concept map visualization should be improved by adding graph visualization algorithms. And the most important task is to integrate the developed tool in already existing student knowledge self-assessment system based on concept maps (Anohina et al., 2006; Anohina et al., 2007). Also, it is planed to create the ontology repository for teachers where subject ontologies in Latvian will be stored.

6 Acknowledgements

This work has been partly supported by the European Social Fund within the National Programme „Support for the carrying out doctoral study programm's and post-doctoral researches" project "Support for the development of doctoral studies at Riga Technical University".

References

Anohina, A., Graudina, V., & Grundspenkis, J. (2006). Intelligent System for Learners' Knowledge Self-Assessment and Process Oriented Knowledge Control Based on Concept Maps and Ontologies. *Annual Proceedings of Vidzeme University College „ICTE in Regional Development"*, 1-8.

Anohina, A., & Grundspenkis, J. (2007). A concept map based intelligent system for adaptive knowledge assessment. *Frontiers in Artificial Intelligence and Applications (Vasilecas, O., Eder, J., Caplinskas, A., Eds.), Vol.155, Databases and Information Systems IV*, 263-276.

Anohina, A., Pozdnakovs, D., & Grundspenkis, J. (2007). Changing the Degree of Task Difficulty in Concept Map Based Assessment System. *Proceedings of the IADIS International Conference "e-Learning 2007"*, 443-450.

Aroyo, L., & Dicheva, D. (2002). Concepts and Ontologies in Web-based Educational Systems. *Proceedings of the International Conference on Computers in Education (ICCE 2002), Vol.2*, 1551-1552.

Bakhtyari, P. (2006). Automatic Feedback Generation – Using Ontology in an Intelligent Tutoring System for both Learner and Author Based on Student Model. *Proceedings of the Eighth International Conference on Enterprise Information Systems: Databases and Information Systems Integration 2006,* 116-123.

Bechhofer, S., van Harmelen, F., Hendler, J., Horrocks I., McGuinness D.L., Patel-Schneider P.F., & Stein L.A. (2004). *OWL Web Ontology Language Reference*. Available online (last visited at 26.06.2008): http://www.w3.org/TR/2004/REC-owl-ref-20040210/

Bloom, B.S. (1956). *Taxonomy of educational objectives. Handbook I: The cognitive domain.* David McKay Co Inc. New York.

Cicognani, A. (2000). Concept Mapping as a Collaborative Tool for Enhanced Online Learning. *Educational Technology & Society 3(3),* 150-158.

Coffey, J.W. (1999). *Institutional memory preservation at NASA Glenn Research Center*. Technical Report, NASA Glenn Research Center, Cleveland, OH.

Croasdell, D.T., Freeman, L.A., & Urbaczewski, A. (2003). Concept Maps for Teaching and Assessment. *Communication of the Association for Information Systems, Vol. 12*, 396-405.

Crowley, R.S., Medvedeva, O., & Jukic, D. (2003). SlideTutor: A model-tracing Intelligent Tutoring System for Teaching Microscopic Diagnosis. *Artificial Intelligence in Education*, 157-164.

Devedzic, V., Debenham, J., & Popovic, D. (2000). Teaching formal languages by an intelligent tutoring system. *Educational Technology & Society, Vol.3, No.2,* 36-49.

Doan, B.-L., & Bourda, Y. (2006). An Educational System Based on Several Ontologies. *Proceedings of the Sixth International Conference on Advanced Learning Technologies 2006 (ICALT'06),* 176-183.

Freeman, L.A. (2004). The Effects OF Concept Maps on Requirements Elicitation and System Models During Information Systems Development. *Proceedings of the First International Conference on Concept Mapping*, 126-133.

Graudina, V. (2008). OWL ontology transformation into concept map. *Scientific Proceedings of Riga Technical University. 5th Series, Computer Science, Applied Computer Science, Vol. 34*, 79-90.

Gruber, T.R. (1992). *ONTOLINGUA: A Mechanism to Support Portable Ontologies.* Technical Report KSL 91-66, Knowledge Systems Laboratory, Stanford University.

Markham, K.M., Mintzes, J.J., & Jones, M.G. (1994). The concept map as a research and evaluation tool: Further evidence of validity. *Journal of Research in Science Teaching. Vol. 31, No. 1,* 91-101.

Na, H.-S., Choi, O.-H., & Lim, J.-E. (2006). A Method for Building Domain Ontologies based on the Transformation of UML Models. *Proceedings of Fourth International Conference on Software Engineering Research, Management and Applications (SERA'06)*, 332-338.

Novak, J.D., & Cañas, A.J. (2006). *The theory underlying concept maps and how to construct them*. Technical Report IHMC CmapTools 2006-01, Institute for Human and Machine Cognition.

Novak, J.D., & Gowin, D.B. (1984). *Learning how to learn*. Cambridge University Press, London.

OMG. *Ontology Definition Metamodel Specification*. Available online (last visited at 26.06.2008): http://omg.org/docs/ptc/06-10-11.pdf

Ruiz-Primo, M.A. (2004). Examining concept maps as an assessment tool. *Proceedings of the First International Conference on Concept Maps.*

van Harmelen, F., Patel-Schneider, P.F., & Horrocks, I. (ed.) (2001). *Reference Description of the DAML-OIL Ontology Markup Language*. Technical Report. Available online (last visited at 7.04.2008): http://www.daml.org/2001/03/reference.html

Vergara Ede, M., & Capuano, N. (2003). DIOGENE: A Training Web Broker for ICT Professionals. *Proceedings of the 2nd International Conference on Multimedia and ICT in Education (m-ICTE 2003), Vol. 1,* 330-334.

Vysniauskas, E., & Nemuraite, L. (2006). Transforming Ontology Representation from OWL to Relational Database. *Information Technology and Control, Vol. 35A, No. 3*, 333-343.

Yin, Y., Vanides, J., Ruiz-Primo, M.A., Ayala C.C., & Shavelson R.J (2005). Comparison of Two Concept – Mapping Techniques: Implications for Scoring, Interpretation, and Use. *Journal of Research in Science Teaching, Vol. 42, No 2*, 168-184.