Ontology Evaluation – Comprising Verification and Validation

Sandra Lovrenčić, Mirko Čubrilo
Faculty of Organization and Informatics
University of Zagreb
Pavlinska 2, 42000 Varaždin, Croatia
{sandra.lovrencic, mirko.cubrilo}@foi.hr

Abstract. *As ontologies became widely used methodology for knowledge representation, as well as foundation for Semantic Web, a question of their evaluation increased even more. Various methods and techniques for ontology evaluation have been already proposed, some concerned with their taxonomy and others with their content. In this paper several ontology evaluation methods are used together with purpose of showing how both important parts of evaluation, verification and validation, can be comprised. The usage of multiple independent evaluation methods ensures development of a consistent and usable ontology.*

Keywords. ontology evaluation, ontology verification, ontology validation, domain ontology, application ontology

1 Introduction

Knowledge representation is an area of research that is being in the focus of interest for decades. With more and more methods and techniques for knowledge representation that have emerged during that time, ontologies [11] are currently one of the most popular and widespread. The first and very well known definition of the term over 15 years ago [13][14] presents an ontology as "a formal explicit specification of a shared conceptualization for a domain of interest", meaning that it "has to be specified in a language that comes with formal semantics". Web ontologies and new technologies of Semantic Web (where ontologies should provide semantic) are the latest research focus, partly because of a rising demand for more knowledge reuse and sharing.

With such a widespread use, the question of ontology evaluation has also increased. Because of that, many various evaluation methods have been developed. Some authors suggest more manual and others more automatic approach. There are methods dealing with ontology taxonomy and those that evaluate their content. Some are dependent on a specific tool or language and some can be used independently. With such a variety of choices, one has to be careful to choose one or more methods that will comprise both evaluation aspects – verification and validation.

This paper tries to show how the use of several evaluation methods together can ensure the quality of an ontology using the example of domain ontology for university studies in Croatia. Both for verification and validation two independent evaluation methods were chosen. Methods are briefly described as well as their application on the example ontology.

2 Evaluation as part of ontology engineering

2.1 Ontology Development

A number of methods and methodologies for ontology development are already proposed [4][5], METHONTOLOGY [11] and On-To-Knowledge [26] being examples of best graded in several methodologies comparisons [11]. Along with those "building from scratch" methodologies, there are other possibilities for ontology development, for example usage of similarities between ontologies [20] or ontology reuse and reengineering [12]. Current work also includes design of a methodology for development of networked ontologies in NeOn project [36].

Almost every methodology offers a specific tool for development, usually with specific evaluation method attached to it. Therefore, a substantial number of tools are available and various plug-ins that resolve conversion among them offer even more possibilities. The choice of a methodology and a tool depends on various factors, such as: a degree of formality, the domain or future extensions of ontology.

2.2 Ontology Evaluation

Similar to information and intelligent systems development process, evaluation of ontologies has become very important, which can be seen from a number of evaluation methods that were already proposed [1][3][8][15][16][25]. Most of ontology development methodologies include also evaluation as their part, either at the end or through the whole ontology development process. Ontology evaluation can be defined as "a technical judgment of the content of the ontology with respect to a frame of reference during every phase and between phases of their lifecycle"[11]. A frame of reference can be requirements specification, competency questions, real world etc. Every evaluation should be consisted of two parts:

- Verification – which "refers to building the ontology correctly, that is, ensuring that its definitions (written in informal or formal language) implement correctly the ontology requirements and competency questions, or function correctly in the real world"[11].
- Validation - which "refers to whether the ontology definitions really model the real world for which the ontology was created. The goal is to prove that the world model (if it exists and is known) is compliant with the world modeled formally"[11].

3 Ontology Evaluation Methods

Some ontology evaluation methods can be applied without a concern about development methodology, but others are connected to individual methods, for example ontology alignment [22], or to certain development languages or tools [23]. The most used and also the most recommended are independent methods, because they can be used for ontology evaluation no matter what methodology or knowledge representation formalism was used for ontology development.

There are many factors that have to be taken into account when evaluating ontology: content, size, possibilities of language and tool used [25], user requirements, simplicity of usage [7][16][21]. Comparison of various methods [2] can make the choice of appropriate method easier. But, since a proper evaluation has to be done through both verification and validation, evaluation methods for both parts have to be used. And to be able to cover all aspects of those two evaluation parts, authors propose using several independent evaluation methods, two for verification and two for validation. Reasons for their choice are:

- independent methods can be used for various types of ontologies developed in different tools and with different knowledge representation formalisms and as such enable easier comparison of evaluation results;
- chosen methods are properly described in literature with examples and are well known and used;
- two evaluation methods for verification have different perspective and therefore can ensure no mistakes in correct domain representation using ontologies;
- in ontology validation, two chosen methods are complementary: content evaluation is more theoretical and application practically proves reference to a real world.

It has to be noted that methods used in this work are applied to small domain ontology. Independent methods, especially those performed manually, would take much more time (probably to much) with large ontologies.

3.1 Ontology Verification

According to a given definition, verification refers to correct representation of a domain and requires checking its definitions. It means that the hierarchy of concepts has to be consistent and also correct according to the real world. For that purpose authors suggest two independent methods: Ontology taxonomy evaluation [8][11] and OntoClean method [11][15][16][27].

3.1.1 Ontology Taxonomy Evaluation

This method is used manually, with ontology checking by a domain expert, according to three main factors [8][11]:

- Inconsistency
  - Circularity errors – a class is defined as a specialization or generalization of itself;
  - Partition errors – improper definition of disjoint classes or incomplete class definition (a class is a subclass of two or more disjoint classes, an instance is object of two or more disjoint classes, an instance is a direct member of a class that has subclasses);
  - Semantic errors – a concept is a subclass of a concept to which it doesn't belong.
- Incompleteness
  - Incomplete concept classification – some domain concepts are missing from taxonomy;
  - Partition errors – undefined relations between some classes (disjoint classes are not defined,

it is not defined that subclasses of a certain class are covering all individuals of that class).

- Redundancy
  - Grammatical redundancy – more than one definition of any class or instance relation is given;
  - Identical formal definition of some classes (with a different name);
  - Identical formal definition of some instances (with a different name).

3.1.2 OntoClean Method

OntoClean, as one of the most known methods, is partially integrated in several tools and there also exist a tool for automatic evaluation of RDF(S) and OWL ontologies according to it. The method is based on four concepts from philosophy, that aim at finding wrong "a is a subclass of *b*" relations in taxonomy. Those four OntoClean method criterions are [11][15][16][27]:

- Rigidity – a concept can be:
  - rigid if and only if it is essential for all of its instances (+R)
  - ant-rigid if and only if it is not essential for all of its instances (~R)
  - non-rigid if and only if it is not essential for some of its instances (-R)
- Identity – a concept can:
  - carry an identity criterion for a property if and only if all its instances can be (re)defined according to it (+I) – otherwise it does not carry an identity criterion (-I)
  - supply an identity criterion for a property if and only if it is rigid and if this property is not inherited by any subsuming property belonging to the ontology (+O) – otherwise it does not supply an identity criterion (-O)
- Unity – a concept:
  - carry unity if there exists a common relation such that all concept instances are always wholes in that relation (+U) - it is inherited;
  - carry anti-unity if there is possibility that all its instances are not wholes in a given relation (~U) – it is inherited;
  - cary non-unity if there is a possibility that some (or all) instances are whole in a given relation, although it doesn't have to be the same for all instances (-U) – it does not have to be inherited.
- Dependency – it is defined between two concepts as:
  - concept B is dependant on concept A if and only if for all instances of concept B must exist some instances of concept A, but they are not a constituent or any other part of B (+D) – it is inherited;
  - concept B is non-dependent on concept A if aforesaid does not apply (-D) - it does not have to be inherited

After assigning criterions, the taxonomy evaluation classes have to satisfy following restrictions:

- all subclasses of anti-rigid class must be anti-rigid;
- rigid class can not be a subclass of non-rigid class;
- a class that supplies an identity criterion also carries an identity criterion;
- all subclasses of a class that carries an identity criterion also must carry an identity criterion;
- all subclasses of a class that carries unity criterion must carry unity criterion;
- all subclasses of a class that carries anti-unity criterion must carry anti-unity criterion
- all subclasses of a dependent class must be dependent;
- classes that does not carry compatible identity and unity criterions must be disjoint.

3.2 Ontology Validation

Since validation refers to the real world, it requires ontology content evaluation. But this also means that it should be checked on the real world example, with concrete application ontology. Therefore, authors suggest two, also independent, methods for validation: an application ontology development and Ontology content evaluation [8][9][10][24].

3.2.1 Application Ontology

Development of an application ontology means that concrete instances from real world or a real world model must be included in the ontology. If the ontology being validated is some kind of a general ontology (as a domain ontology in our example), it should be adjusted (if necessary) to a concrete example and populated with instances. Adjustment means that some classes will be described with more attributes or/and that several new subclasses will be added because of specifics of a concrete domain case. Domain expert and users should then evaluate obtained result.

If the given ontology already is application ontology, this part of validation can only include ontology checking from a domain experts and users against a real world.

3.2.2 Ontology Content Evaluation

Content evaluation is also performed manually, with ontology checking by a domain expert, according to [8][9][10][24]:

- Consistency - refers to whether it is possible to obtain contradictory conclusions from valid input

definitions. A given definition is consistent if and only if each definition is individually consistent (informal and formal definition aren't contradictory according to the real world and to each other) and no contradictory knowledge can be inferred from all definitions and axioms;
- Completeness - incompleteness is a fundamental problem in ontologies. Ontology is complete if all that is supposed to be in the ontology is explicitly stated in it or can be inferred and if each definition is complete (whether it defines the whole domain explicitly or knowledge can be inferred from other definitions and axioms);
- Conciseness - ontology is concise if: it does not contain any unnecessary or useless definitions and explicit or implicit redundancies between definitions of terms;
- Expandability – ontology is easily expandable if there is no need to alter a set of well-defined definitions in the case of adding new definitions and new knowledge to existing ones;
- Sensitiveness – ontology is not sensitive if small changes in definition doesn't alter a set of well-defined concepts.

4 Ontology Example Development

4.1 Ontology Domain

At the very beginning of the domain description it should be explained that studies at universities in Croatia are divided into university and occupational studies. The subject domain of this paper represents only university studies and all concepts and their properties and relations in ontology are directly or indirectly connected to them. The current concept of studies was new when the development started and therefore very interesting. The domain knowledge was easily gathered from all important sources [28][29][30][31][32][33][34][35], but only the law could have been considered for domain modeling. Namely, the aim was to represent studies at the level of the whole country and statutes of six existing Croatian universities didn't have any new concepts equal for all universities that were not already defined by law. The main groups of obtained concepts were:

- types of studies;
- conduction places;
- teaching participants;
- courses;
- enrolment requirements.

Those five groups of concepts were connected to the root concept called Thing, since they all together describe a subject domain. The domain model was developed in the description logics (DL), which are knowledge representation formalisms that are a decisive part of first order logic [1][17]. The model was implemented in a Protégé-OWL tool. Since this tool enables automatic taxonomy classification with Racer tool [38], which is also based on DL, this step was done too. Model development and implementation are in detail described in [19]. The final concept hierarchy created according to the model and after classification is as follows:

- Thing
  - Study
    - Occupational_study
    - University_study
      - Undergraduate_study
      - Graduate_study
      - Postgraduate_study
        - Postgraduate_doctoral_study
        - Postgraduate_specialist_study
  - Course
  - Enrolment_requirements
    - Study_enrolment_requirements
    - Year_enrolment_requirements
  - Conduction_place
    - Higher_education_institution
      - University
      - Polytechnic
      - School_of_profess_higher_education
    - Science_institute
    - Constituent_unit
      - Faculty
      - Academy of art
      - University_department
      - University_institute
      - Other_constituent_unit
    - Other_institution
  - Teaching_participant
    - Teacher
      - Scientific_educational_title
      - Artistic_educational_title
      - Educational_title
      - Scientific_title
      - Associate_title
      - Professional_title
      - Honorary_title
      - Visiting_teacher
    - Student

4.2 Ontology Implementation

Ontology was finally developed in Protégé-Frames. This tool was chosen because it is one of the most known ontology development and knowledge representation tools that is being constantly developed for more than 20 years [37].

Domain model was easily translated from Protégé-OWL to Protégé-Frames. Concepts in the ontology were already organized into class hierarchy. To all classes were attached their attributes (called slots) and their restrictions. In Protégé, restrictions were made in slot editor, but also as axioms expressed in formal logic using PAL – Protégé Axiom Language. The syntax of the language is a variant of Knowledge Interchange Format (KIF) [6] and it has constraint-checking engine that can be run against the knowledge base to find frames that violate constraints. Ontology implementation is in detail described in [18].

5 Ontology Example Evaluation

All four evaluation methods described above were applied to evaluate university studies ontology. Evaluation results are presented below.

5.1 Verification

5.1.1 Ontology Taxonomy Evaluation

The method gave following results:

- Inconsistency
  - Circularity errors – no errors;
  - Partition errors – no errors during final evaluation, classes Faculty and Academy_of_art, initially subclasses of University, were transferred to be subclasses of Constituent_unit after the model was developed and classified because University can have concrete individuals;
  - Semantic errors – no errors.
- Incompleteness
  - Incomplete concept classification – all concepts from knowledge sources were included into ontology;
  - Partition errors – no errors.
- Redundancy
  - Grammatical redundancy – no errors;
  - Identical formal definition of some classes (with a different name) – no errors;
  - Identical formal definition of some instances (with a different name) – in domain ontology there are no instances.

5.1.2 OntoClean Method

After all criterions were carefully attached to each class, values were implemented in Protégé using a special plug-in that is developed for OntoClean. All restrictions that have to be complied, except the last one, were checked automatically. They can also be checked manually, as was necessary for the last restriction. Both checks showed that all restrictions were satisfied, which was expected after taxonomy evaluation. Fig. 1 shows obtained results.

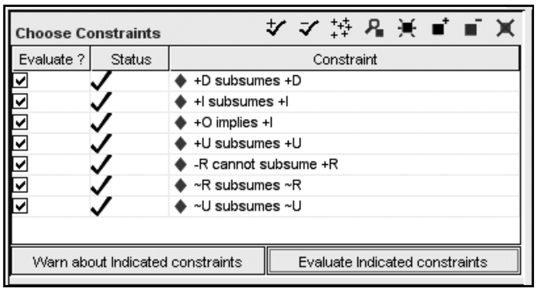

Figure 1. OntoClean method results

5.2 Validation

5.2.1 Application Ontology

Since the original ontology represented domain ontology without instances, concrete ontology for university studies at Faculty of organization and informatics in Varaždin, Croatia was created. For that purpose several adjustments through the whole ontology had to be made both to the:

- Formal model
  - 4 new definitions for 4 new classes were created;
  - 5 existing definitions were altered in a way that new classes are added into definitions where necessary.
- Ontology
  - 4 new classes with attributes were added to the taxonomy at the lowest hierarchy level;
  - 23 new attributes were added to existing classes;
  - 1 class changed from concrete (can have individuals) to abstract (can not have individuals) because of new classes;
  - 3 new axioms were added.

Into adjusted ontology concrete instances were successfully added. Domain expert, a faculty professor most involved in new studies concept design helped during the development and approved obtained results.

5.2.2 Ontology Content Evaluation

Content evaluation was performed both on domain and application ontology, with the following results:

- Consistency – ontology is consistent individually (regarding informal and formal definitions) and no contradictory knowledge can be inferred from all definitions and axioms:
- Completeness – regarding information from knowledge sources, ontology is complete, but since legal documents possibly have loopholes, there is a possibility that even application ontology isn't complete.
- Conciseness – ontology does not contain any unnecessary concepts (for example, Teaching_participant is a direct subclass of Thing, class Person is omitted). On the other side, complete and detailed definition of a formal model has caused implicit redundancies (some concept definitions can be inferred from others). Therefore ontology isn't concise.
- Expandability – development of application ontology showed that concept hierarchy doesn't have to be significantly changed. Alteration of model definitions didn't influence on their meaning nor new axioms didn't influence on other axioms and classes. Most of new restrictions are

made very easily with new attributes. Therefore ontology is easily expandable.

- Sensitiveness – as already stated in expandability, small changes in definitions of several concepts didn't change the set of well-defined concepts, so ontology isn't sensitive.

6 Conclusion

Ontology evaluation, for example given in this paper was performed using methods that cover both verification and validation. For each of those evaluation parts, two independent methods were chosen ensuring the evaluation completeness. The justification of double evaluation can be further discussed.

Evaluation showed that taxonomy of small ontologies can be created with only few mistakes, but with the content that may not be the case. The omnipresent problem of ontology incompleteness is almost always present. Most part of the problem are attributes and relations that are not always explicitly denoted in the domain, as are main concepts. Also, the advantage of strict conciseness of ontology model on one side and full formal definitions on the other can be explored further.

All presented methods are all or in most part performed manually. For a small ontology, as presented in this paper, this doesn't represent an obstacle. But with larger ontologies automatic methods could be more appropriate because of the time involved. Of course, some parts of evaluation (as ontology completeness) cannot be easily performed automatically. Whatever the choice, ontology engineers should always check whether it comprises both verification and validation.

References

[1] Baader F, Calvanese D, McGuinness D, Nardi D, Patel-Schneider P (eds.): **The Description Logic Handbook**, Cambridge University Press, Cambridge, UK, 2003.

[2] Brank J, Grobelnik M, Mladenič D: **A Survey of Ontology Evaluation Techniques**, Conference on Data Mining and Data Warehouses 17th October, Ljubljana, 2005, available at `http://kt.ijs.si/dunja/sikdd2005/Papers/BrankEvaluationSiKDD2005.pdf`, Accessed: 20th December 2005.

[3] Damjanović V, Gašević D, Devedžić, V B: **Analysis of Ontology Validation Possibilities**, Proceedings of the 4th International Workshop on Computational Intelligence and Information Technologies 13th October, Niš, Serbia and Montenegro, 2003, pp. 47-50.

[4] Fernández-López M, Gómez-Pérez A: **Overview and analysis of methodologies for building ontologies**, The Knowledge Engineering Review, 17(2), Cambridge, UK, 2002, pp. 129-156.

[5] Fernández-López M et. al.: **A survey on methodologies for developing, maintaining, integrating, evaluating and reengineering ontologies**, 2002, available at `http://www.aifb.uni-karlsruhe.de/WBS/ysu/publications/OntoWeb_Del_1-4.pdf`, Accessed: 05th February 2005.

[6] Genesereth M R: **Knowledge Interchange Format Specification**, 1995., available at `http://logic.stanford.edu/kif/specification.html`, Accessed: 20th October 2004.

[7] Giboin A, Gandon F, Corby O, Dieng R: **Assessment of Ontology-based Tools: A Step Towards Systemizing the Scenario Approach**, Proceedings of EON2002: Evaluation of Ontology-based Tools Workshop at the 13th International Conference on Knowledge Engineering and Knowledge Management (EKAW 2002) 30th September, Siguenza, Spain, 2002, pp. 63-73.

[8] Gómez-Pérez A: **Evaluation of Ontologies**, International Journal of Intelligent Systems, 16, Chichester, UK, 2001, pp. 391-409.

[9] Gómez-Pérez A: **Guidelines to Verify Completeness and Consistency in Ontologies**, Critical Technology: Proceedings of the 3rd world congress on expert systems 5th – 9th February, Seoul, South Korea, 1996, pp. 901-908.

[10] Gómez-Pérez A: **Towards a Framework to Verify Knowledge Sharing Technology**, Expert Systems With Applications, 11(4), New York, USA, 1996, pp. 519-529.

[11] Gómez-Pérez A, Fernández-López M, Chorcho O: **Ontological Engineering**, Springer-Verlag, London, UK, 2004.

[12] Gómez-Pérez A, Rojas-Amaja M D: **Ontological Reengineering for Reuse**, Proceedings of the 11th European Workshop on Knowledge Acquisition, Modeling and Management 26th – 29th May, Dagstuhl Castle, Germany, 1999, pp. 139-156. (Lecture Notes in Computer Science)

[13] Gruber T R: **A translation Approach to Portable Ontology Specifications**, Knowledge Acquisition, 5(2), London, UK, 1993, pp. 199-220.

[14] Gruber T R: **Toward Principles for the Design of Ontologies Used for Knowledge Sharing**, in Guarino N and Poli R (eds.): Formal ontology in conceptual analysis and knowledge representation, Kluwer Academic Publishers, 1993, available at `ftp://ftp.ksl.stanford.edu/pub/KSL_Reports/KSL-93-04.ps.gz`, Accessed: 20th November 2005.

[15] Guarino N, Welty C: **A Formal Ontology of Properties**, Proceedings of the 12th European Workshop on Knowledge Acquisition, Modeling and Management 2nd – 6th October, Juan-les-Pins, France, 2000, pp. 97-112. (Lecture Notes In Computer Science)

[16] Hartmann J et. al.: **D1.2.3 Methods for ontology evaluation**, 2005, available at `http://knowledgeweb.semanticweb.org/semanticportal/servlet/download?ontology=Documentation+Ontology&concept=Deliverable&instanceSet=kweb&instance=D1.2.3%3A+Methods+for+ontology+evaluation&attribute=Online+PDF+Version &value=KWeb-Del-1.2.3-Revised-v1.3.pdf`, Accessed: 20th December 2005.

[17] Horrocks I: **Description Logics in Ontology Applications**, Proceedings of the 9th International

Conference on Automated Reasoning with Analytic Tableaux and Related Methods (TABLEAUX 2005) 14th – 17th September, Koblenz, Germany, 2005, pp. 2-13. (Lecture Notes in Artificial Intelligence)

[18] Lovrenčić S, Čubrilo M: **Ontologies in the Higher Education Domain**, Proceedings of the 18th International Conference on Information and Intelligent Systems (IIS 2007) 20th – 22nd September, Varaždin, Croatia, 2007.

[19] Lovrenčić S, Čubrilo M: **University Studies Ontology – Domain Modeling**, Proceedings of a 11th IEEE International Conference on Intelligent and Engineering Systems (INES 2007) 39th June – 1st July, Budapest, Hungary, 2007, pp. 55-58.

[20] Maedche A, Staab S: **Measuring Similarity between Ontologies**, Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management - Ontologies and the Semantic Web (EKAW 2002) 1st – 4th October, Siguenza, Spain, 2002, pp. 251-263. (Lecture Notes In Computer Science)

[21] Noy N F, Guha R, Musen M A: **User ratings of ontologies: Who will rate the raters?**, Proceedings of the AAAI 2005 Spring Symposium on Knowledge Collection from Volunteer Contributors 21st – 23ed March, Stanford, USA, 2005, pp. 56-63.

[22] Noy N F, Musen M A: **Evaluating Ontology-Mapping Tools: Requirements and Experience**, Proceedings of EON2002: Evaluation of Ontology-based Tools Workshop at the 13th International Conference on Knowledge Engineering and Knowledge Management (EKAW 2002) 30th September, Siguenza, Spain, 2002, pp. 1-14.

[23] Seipel D, Baumeister J: **Declarative Methods for the Evaluation of Ontologies**, Künstliche Intelligenz, 4, Bremen, Germany, 2004, pp. 51-57.

[24] Staab S, Studer R (eds.): **Handbook on Ontologies**, Springer-Verlag, Berlin, Germany, 2004.

[25] Sure Y, Gómez-Pérez A, Daelemans W, Reinberger M-L, Guarino N, Noy N F: **Why Evaluate Ontology Technologies? Because it Works!**, IEEE Intelligent Systems, 19(4), Washington, USA, 2004, pp. 74-81.

[26] Sure Y, Studer R: **On-To-Knowledge Methodology – Final Version**, 2002, available at `http://www.aifb.uni-karlsruhe.de/WBS/ysu/publications/OTK-D18_v1-0.pdf`, Accessed: 05th February 2005.

[27] Welty C, Guarino N: **Supporting ontological analysis of taxonomic relationships**, Data & Knowledge Engineering, 39, New York, USA, 2001, pp. 51-74.

[28] ...: **Statut Sveučilišta u Dubrovniku** (The Statute of the University of Dubrovnik), Sveučilište u Dubrovniku, Dubrovnik, Croatia, 2006.

[29] ...: **Statut Sveučilišta u Osijeku** (The Statute of the University of Osijek), Sveučilište u Osijeku, Osijek, Croatia, 2005.

[30] ...: **Statut Sveučilišta u Rijeci** (The Statute of the University of Rijeka), Sveučilište u Rijeci, Rijeka, Croatia, 2004.

[31] ...: **Statut Sveučilišta u Splitu** (The Statute of the University of Split), Sveučilište u Splitu, Split, Croatia, 2005.

[32] ...: **Statut Sveučilišta u Zadru** (The Statute of the University of Zadar), Sveučilište u Zadru, Zadar, Croatia, 2005.

[33] ...: **Statut Sveučilišta u Zagrebu** (The Statute of the University of Zagreb), Sveučilište u Zagrebu, Zagreb, Croatia, 2005.

[34] ...: **Zakon o izmjenama i dopunama Zakona o znanstvenoj djelatnosti i visokom obrazovanju**, (The law of changes and additions to the Science and higher education legislation act) Narodne novine, No. 105/2004, Zagreb, Croatia, 2004.

[35] ...: **Zakon o znanstvenoj djelatnosti i visokom obrazovanju** (Science and higher education legislation act), Narodne novine, No. 123/2003, Zagreb, Croatia, 2003.

[36] **NeOn**, http://www.neon-project.org

[37] **Protégé**, http://protege.stanford.edu

[38] **Racer**, http://www.racer-systems.com, Copyright © 2004, 2005 by Racer Systems GmbH & Co. KG