INFLUENCE AND SELECTION OF BASIC CONCEPTS ON ONTOLOGY DESIGN

Tomasz Boinski, Piotr Orlowski, Piotr Szpryngier and Henryk Krawczyk
Faculty of Electronics, Telecommunications and Informatics, Gdansk University of Technology
11/12 Gabriela Narutowicza Street, Gdansk, Poland

Keywords: Ontology, Set of basic concepts, Ontology learning.

Abstract: Ontologies as entities representing individual point of view on surrounding world introduce heterogeneity to knowledge representation. Common set of core concepts can introduce similarity big enough for further interoperability between ontologies. In this paper an experiment is presented proving that despite differences in detail ontologies stay similar in regard of core concepts. During the experiments NOIA methodology enhanced by OCS methodology was used to create ontologies from three significantly different sources of knowledge about risk management.

1 INTRODUCTION

Ontology creation is a complex task dependent on methodology and a set of basic concepts. Ontologies presenting different points of view limit interoperability, making mapping or merging a tedious task.

To be usable by a wide range of recipients the created ontology needs to represent a common view of the described problem. Humans, however tend to have a distinct way of perceiving the surrounding reality. Different sets of basic concepts can be used and utilized. Furthermore a common set of basic concepts not always means a common set of basic definitions. Even slight differences can lead to potentially different ontologies. There are some upper ontologies proposed (Niles and Pease, 2001) (Masolo et al., 2003) but it's highly unlikely that a global agreement will be met upon a common set of concepts or the shape of an ontology. Without such agreement upon basic concepts and their definitions, interactions between systems using different ontologies will be difficult - a need for costly manual mapping arises.

The purpose of this paper is to test whether using distinct sets of definitions for chosen sets of basic concepts can provide ontologies similar enough to interoperate with each other. In other words, how the choice of a knowledge source implies the final structure of the ontology.

Objectives for this test were as follow:

1. To choose the ontology domain and three different sources of knowledge.
2. To build three ontologies using chosen knowledge sources.
3. To evaluate prepared ontologies.
4. To compare result ontologies and conclude the research.

The following section will present the initial assumptions. Next, in section 3, methodology and groupwork model used for ontology creation will be described. In section 4 the basic set of concepts and their definitions are described. In section 5 the process of creating ontologies and achieved results are presented. Section 6 shows achieved results.

2 INITIAL ASSUMPTIONS

2.1 Ontology Domain

Choice of a vague domain of concepts was needed to allow use of more diversified knowledge sources. The risk has many, often contrary, definitions i.e. (Waste, 2006) (Hall and Hulett, 2002) (Knight, 2002) and that was the main reason of choosing it as the ontology domain.

2.2 The Ontology Purpose and Scope

The prepared ontology has a strictly research purpose. It should answer the question, what are the main rela-

Boinski T., Orlowski P., Szpryngier P. and Krawczyk H..
INFLUENCE AND SELECTION OF BASIC CONCEPTS ON ONTOLOGY DESIGN.
DOI: 10.5220/0003076103640369
In *Proceedings of the International Conference on Knowledge Engineering and Ontology Development* (KEOD-2010), pages 364-369
ISBN: 978-989-8425-29-4

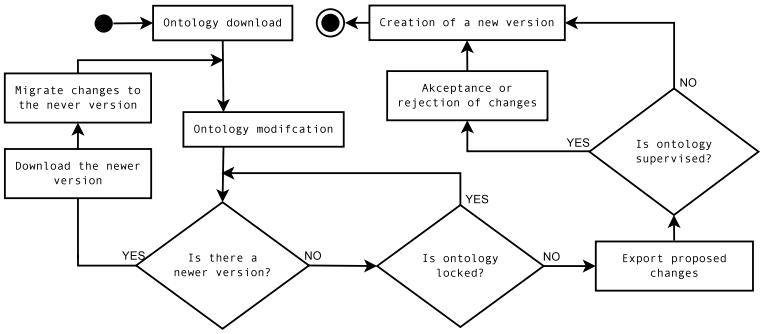

Figure 1: Process of creating new version of an ontology in OCS.

tions between the concepts of risk, threat, vulnerability and asset. These terms appear in most risk definitions.

2.3 Knowledge Sources

Basing on a search of the literature, three sources of knowledge were chosen:

- NIST Glossary of Key Information Security Terms (Kissel, 2006),
- ENISA Risk Management Glossary (Enisa, 2010),
- Software Engineering, Ian Sommerville (Sommerville, 2006).

Two of them are glossaries prepared by recognized organizations: one American, NIST - National Institute of Standards and Technology and one European, ENISA - European Network and information Security Agency. The third one is a recognized software engineering university coursebook from The United Kingdom, but it also contain definitions of used terms. All are texts written in natural language and describing security domains. NIST glossary contains definitions used in USA and gives us a view to terminology used in America. ENISA Glossary was prepared for European Union Institutions and Member States. Consequently it presents terminology used in Europe and often utilises ISO (International Organization for Standardization) definitions. The book written by Ian Sommerville presents risk terminology in a more objective way, according to concepts used by software engineers.

2.4 Used Groupwork Model and Software

Ontologies have been created using a methodology initially designed for Ontology Creation System (OCS) (Boinski et al., 2009) developed by The Department of Computer Architecture of Gdansk University of Technology. Opposed to methodology available in Protegeeditor (Gennari et al., 2002) (Noy et al., 2000) with Collaborative Protege (Tudorache et al., 2008) extension, this methodology ensures that the creator of the ontology will retain control over it. Whereas Protege resolves conflicts by means of voting, OCS methodology always leaves the final word to the ontology owner.

Any registered user can propose some changes in any ontology. A privileged user, i.e. ontology creator or an expert designated by him, can accept or reject proposed changes. After that, a new version is created and accepted changes are available for other developers. The whole process is showed in Figure 1. When an ontology is public and its development is not restricted in any way, the process of submitting propositions of changes can be combined with creating a new version.

For the purpose of this paper we distinguish a security expert, who was assigned the role of an ontology owner and all changes needed to be accepted by him.

Table 1: Aggregated results of ontology comparison using Falcon-OA (Jian et al., 2005).

	ENISA	NIST	Sommerville
ENISA	X	26	17
NIST	26	X	18
Sommerville	17	18	X

3 RESEARCH METHODOLOGY

3.1 Ontology Learning

There are many methodologies of learning ontologies from natural text, for example (McGuiness and

Table 2: Similarity between ENISA and NIST based ontologies.

No.	Concept name in ENISA	Concept name in NIST	Similarity
1	Group	Group	1.0
2	Operation	Operation	0.9989048162438352
3	Safeguard	Safeguard	0.9984604434491676
4	Mission	Mission	0.9979042801363289
5	Denial_Of_Services	Denial_Of_Service	0.9900581108148236
6	Attack	Attack	0.9805473496555192
7	Weakness	Weakness	0.9641696140783707
8	Harm	Harm	0.9203099356002691
9	Circumstance	Circumstance	0.9177295789689781
10	Potential	Potential	0.9140700859364972
11	Threat	Threat	0.8307676361865943
12	Event	Event	0.811927478336268
13	Modification_Of_Data	Unauthorized_Modification_Of_Data	0.8098784893405487
14	Implementation_Error	Implementation	0.797691170284476
15	Risk	Risk	0.7695468416659597
16	Vulnerability	Vulnerability	0.756619672104696
17	Procedure	Security_Procedure	0.7487199546707506
18	Application	Major_Application	0.7344763940853262
19	Control	Security_Control	0.713245540577225
20	Unauthorized_Access	Unauthorozed_Access	0.7084168873642644
21	Destruction	Unauthorized_Destruction	0.7079624012576695
22	exploits	exploits	0.999738325227409
23	isMemberOf	isMemberOf	0.9994788690128461
24	hasMember	hasMember	0.9992265192460275
25	causes	causes	0.9898447200777993
26	isPotentialOf	isPotentialThat	0.9075646311625767
27	hasPotentialToImpact	hasPotentialTo	0.7669824888017037

Table 3: Similarity between ENISA and Sommerville based ontologies.

No.	Concept name in ENISA	Concept name in Sommerville	Similarity
1	Value	Value	1.0
2	Weakness	Weakness	0.9564709904052531
3	Potential	Potential	0.9271158225611422
4	Harm	Harm	0.9209783867451191
5	Circumstance	Circumstance	0.9171607954808378
6	Attack	Attack	0.8580363426992699
7	Control	Control	0.8539265029954892
8	Asset	Asset	0.8204264111517303
9	Computer_system	Computer_Based_System	0.7963994582890108
10	Information_Resource	Resource	0.7959453114871415
11	Threat	Threat	0.7820141153736444
12	Event	Event	0.7815727930842786
13	Risk	Risk	0.758768058194784
14	Vulnerability	Vulnerability	0.73883024261744
15	reduces	reduces	0.9997731663396932
16	hasValue	hasValue	0.9982319005180453
17	exploits	canBeExploited	0.777640801540454

Noy, 2005) or (Fernandez et al., 1997). The modified "Noy and McGuiness" approach was used here. Original "Noy and McGuiness" Knowledge Engineering Methodology consist of 7 steps:

1. To determine the domain and the scope.
2. To consider reusing of existing ontologies.
3. To enumerate important terms.
4. To define classes and class hierarchy.
5. To define properties of classes.

Table 4: Similarity between NIST and Sommerville based ontologies.

No.	Concept name in NIST	Concept name in Sommerville	Similarity
1	Potential	Potential	1.0
2	Protective_Measure	Protective_Measure	0.9997987609025334
3	Circumstance	Circumstance	0.9997494477352709
4	Weakness	Weakness	0.9995264158511349
5	Harm	Harm	0.9988494092141691
6	Information_System_Vulnerability	System_Vulnerability	0.8857014523683819
7	Attack	Attack	0.8825122910116765
8	Event	Event	0.8683697978162398
9	Loss_Of_Information	Loss	0.8139334750047373
10	Vulnerability	Vulnerability	0.8110371584661011
11	Asset	Asset	0.7870205803749836
12	System	System	0.7637258019185644
13	Risk	Risk	0.7144767829186948
14	Threat	Threat	0.7110288987415302
15	concerns	concerns	0.999822352134665
16	canBeExploited	canBeExploited	0.9967319547095099
17	resultsFrom	resultsIn	0.8593064400982773
18	exploits	isExploitationOF	0.7494414710908577

6. To define properties restrictions.
7. To create instances.

Steps 2 and 7 were not performed. As the knowledge source in step 3 we used 3 different documents. Terms were taken from different definitions of Risk, Threat, Vulnerability, Asset and Safeguard. Step 6 was extended by adding definitions to classes (to change these classes into protege defined classes), according to definitions found in glossaries. All steps were enhanced by groupwork possibilities introduced by OCS methodology.

3.2 Ontology Evaluation

Evaluation was performed with use of advice given in (Staab and Studer, 2009). By evaluation we mean *verification* and *validation* of our ontology. The *verification* answers the question "did we build ontology in a correct way?". The *validation* answers the question "did we build the proper ontology?". The following quality criteria were considered during evaluation phase:

- accuracy - does the ontology present knowledge given in knowledge source?
- clarity - is the ontology understandable? is it documented?
- completeness - does it cover the domain of interest?
- consistency - does it match the specification?

Chosen methodologies made both the ontology owner and the participants responsible for the evaluation of the ontology. OCS methodology required double checking of created ontologies asking the aforementioned questions multiple times - once by normal users suggesting changes in ontology, and later by the ontology owner during the process of changes acceptance. Using Protege and Pellet (the reasoner plug-in used by Protege) provided the proper structure of OWL file.

4 ACHIEVED RESULTS

Three ontologies were constructed. Their inferred hirarchies are depicted on Figure 2. Final ontologies had different number of classes. ENISA based ontology had 43 classes (Figure 2 a), Sommerville based had 38 classes (Figure 2 b) and NIST based had 71 classes (Figure 2 c). They were compared using Falcon-AO (Jian et al., 2005) Ontology Matching tool. Falcon is based on linguistic matching for ontologies and uses promising (Euzenat and Shvaiko, 2007) graph modeling algorithms (GMO) with support of WordNet technology. Similarity of two entities from two ontologies comes from the combination of similarities of involved statements (triples) taking the two entities as the same role (subject, predicate, object) in the triples, while the similarity of two statements comes from the accumulation of similarities of involved entities of the same role in the two statements being compared (Hu et al., 2005). Compacted results are presented in Table 1. Values in the table represent the number of common concepts found in ontologies based on knowledge sources designated by

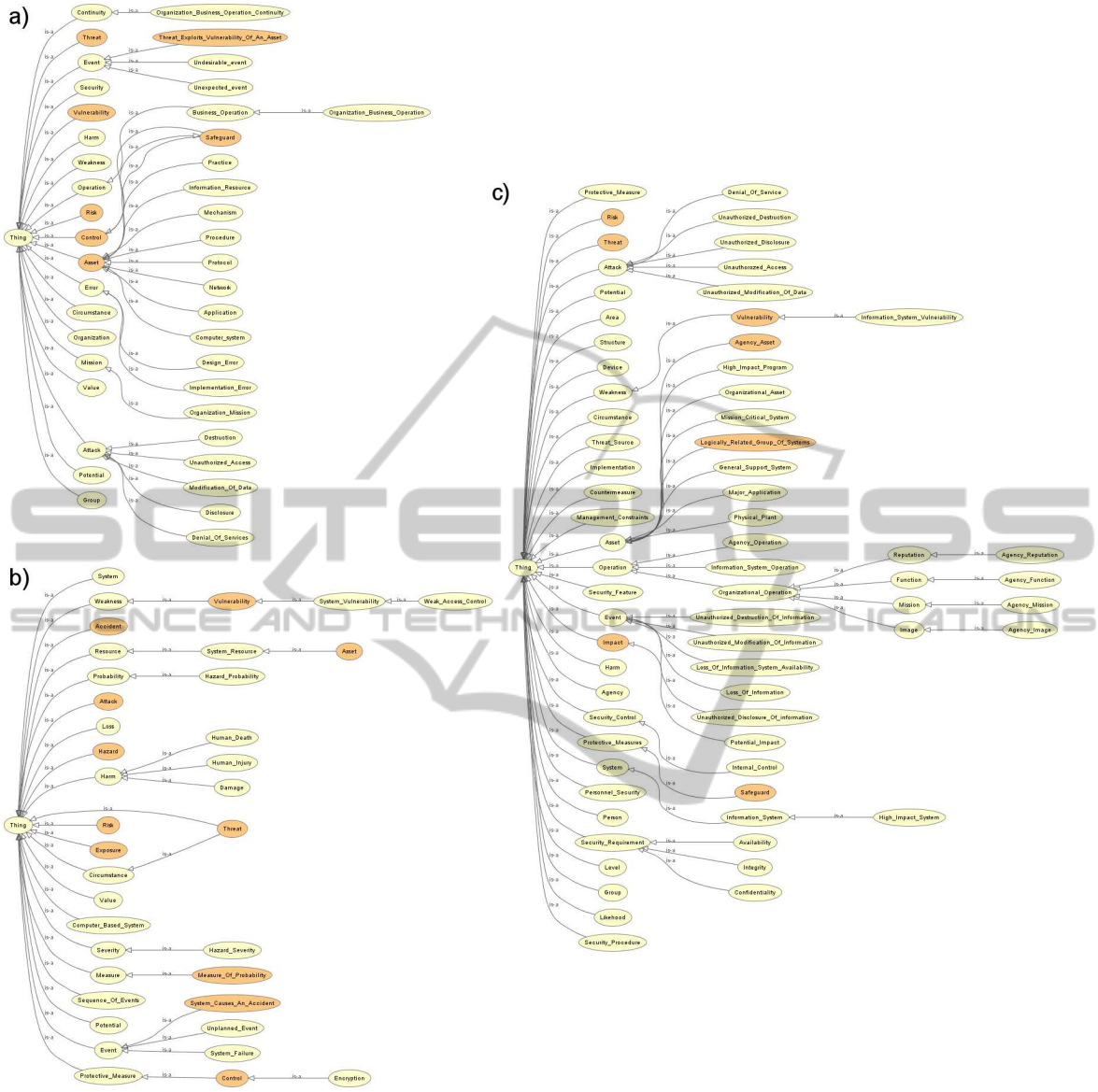

Figure 2: ENISA (a), Sommerville (b) and NIST (c) based ontology.

row and column.

Full results (Tables 2, 3 and 4) show that main concepts are very similar. Differences occur mainly in leaf classes differentiating ontologies in details but not in main elements. Five core concepts (Risk, Threat, Vulnerability, Asset and Safeguard), and some other meaningful concepts, i.e. Attack, Weakness, Event and Harm were found similar in over 70% of all created ontologies.

5 CONCLUSIONS

Performed experiments show that despite differences in details, ontologies proved to be similar in regard of core concepts. Choosing a common set of basic entities enables users to design their own ontologies, yet creating a common base for interoperability at execution time. With the introduction of semantic dictionaries like WordNet (Fellbaum et al., 1998) such undertakings became possible and widely accepted. Common sets of core concepts defined by such dictionaries allow merging or mapping of ontologies on reasoning

time, using knowledge from both ontologies. Concepts from chosen core sets are used as a bridge between those ontologies, opening new opportunities in knowledge integration and creating new possibilities when interoperating with other parties.

As a next step, research on ontology merging and integration will be performed examining new possibilities emerging from usage of a common set of core concepts.

REFERENCES

Boinski, T., Budnik, L., Jakowski, A., Mrozinski, J., and Mazurkiewicz, K. (2009). OCS – Domain Oriented Ontology Creation System. In *SMI'09, 4th International Conference 'Congress of Young IT Scientists'*. HARD Olsztyn.

Enisa (2010). Enisa: a European Union Agency - Glossary of Risk Management. *http://www.enisa.europa.eu/act/rm/cr/risk-management-inventory/glossary*.

Euzenat, J. and Shvaiko, P. (2007). *Ontology matching*. Springer-Verlag New York Inc.

Fellbaum, C. et al. (1998). *WordNet: An electronic lexical database*. MIT press Cambridge, MA.

Fernandez, M., Gomez-Perez, A., and Juristo, N. (1997). Methontology: from ontological art towards ontological engineering. In *Proceedings of the AAAI97 Spring Symposium Series on Ontological Engineering*, pages 33–40.

Gennari, J. H., Musen, M. A., Fergerson, R. W., Grosso, W. E., Crubzy, M., Eriksson, H., Noy, N. F., and Tu, S. W. (2002). *The evolution of Protege: An environment for knowledge-based systems development*. Stanford Medical Institute, Stanford.

Hall, D. and Hulett, D. (2002). Universal risk project, final report. *Risk Special Interest Group, PMI*.

Hu, W., Jian, N., Qu, Y., and Wang, Y. (2005). Gmo: A graph matching for ontologies. In *Integrating Ontologies Workshop Proceedings*, page 41. Citeseer.

Jian, N., Hu, W., Cheng, G., and Qu, Y. (2005). Falcon-AO: Aligning ontologies with Falcon. In *Integrating Ontologies Workshop Proceedings*. Citeseer.

Kissel, R. (2006). Glossary of key information security terms. *Glossary, National Institute of Standards and Technology, US Department of Commerce*.

Knight, F. (2002). *Risk, uncertainty and profit*. Beard Books Inc.

Masolo, C., Borgo, S., Gangemi, A., Guarino, N., and Oltramari, A. (2003). *WonderWeb Deliverable D18*. Laboratory For Applied Ontology - ISTC-CNR, Trento, Italy.

McGuiness, D. and Noy, N. (2005). Ontology development 101: a guide to creating your first ontology. *Universidad de Stanford*.

Niles, I. and Pease, A. (2001). Towards a Standard Upper Ontology. In *Formal Ontology in Information Systems, Proceedings of the international conference on Formal Ontology in Information Systems*, Ogunquit, Maine, USA.

Noy, N. F., Fergerson, R. W., and Musen, M. A. (2000). The knowledge model of Protege-2000: Combining interoperability and flexibility. In *Lecture Notes in Computer Science*. Springer-Verlag.

Sommerville, I. (2006). Software Engineering. 8th. *Harlow, UK: Addison-Wesley*.

Staab, S. and Studer, R. (2009). *Handbook on ontologies, Ontology Validation*. Springer Verlag.

Tudorache, T., Noy, N. F., Tu, S. W., and Musen, M. A. (2008). Supporting collaborative ontology development in Protege. In *Seventh International Semantic Web Conference*, Karlsruhe, Germany.

Waste, R. (2006). IAEA Safety Glossary.