

Supporting Domain Model Authoring

Matej KLOSKA*

*Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
matej.kloska@gmail.com*

Nowadays, we can see the Web as a huge repository of information. Quantity of information is such huge that we talk in general about the overload of information. In order to treat people to access information on the Web effectively, several approaches were created, including for example adaptation (content, presentation / forms, etc.).

Adaptation is now almost everywhere on the Web, in email, in the web; in tablets, in mobile phones. Information relevant for adaptation is stored in adaptation and domain models, which are often represented by graphs. Manual creation of such models is a difficult task and one of the drawbacks of adaptive systems. The basic problems which may be encountered when working with models:

- unfamiliarity with domain model,
- unawareness of relationships in this model,
- flooding the user with automatically generated entities,
- navigation cumbersome,
- non-intuitive user interface for working with models.

These problems are also supported by empirical study [3], especially complexity of the ontology development process from the perspective of authors who are typically unaware of ontology existence and its relevancy altogether.

The aim of our work is to propose a method for supporting domain model authoring. The output of our method should be infrastructure for model management and framework for user friendly interface supporting all above mentioned issues. In our work we are focusing on lightweight ontologies in form of terms, concepts and relations.

There are several approaches to support domain model authoring. Most of them are focused on partial tasks such as identifying model changes, versioning, specification of transformation rules between different ontologies versions, merging, annotations and model partitioning [1]. We are primarily concerned with model generation and maintenance efforts consisting primarily of the following scenarios [2]:

* Supervisor: Marián Šimko, Institute of Informatics and Software Engineering

- versioning of models with the respect to navigation in versions,
- merging multiple models into one aggregate,
- annotation and partition of models.

All of these scenarios are observed to be essential not only to the ontology/model developers but also to software developers. Main subtask for versioning and merging of software codes as well as ontologies is to yield list of differences / steps required to be applied to source version in order to get target version. Software code is represented in form on texts, easy to differentiate – create lists of different lines. On the other hand, differentiation of two ontologies cannot be text based because two ontologies can be the same conceptually but have different text representation. This fact is also depicted in Figure 1.

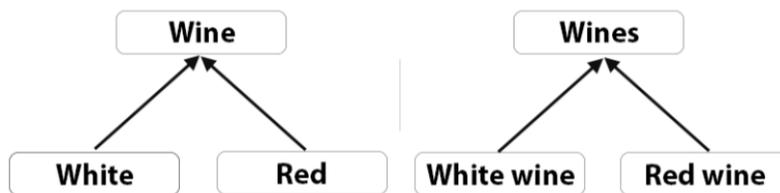


Figure 1. Conceptually same but lexically different models.

Acknowledgement. This work was partially supported by the Cultural and Educational Grant Agency of the Slovak Republic, grant No. 009STU-4/2014.

References

- [1] Noy, N.F, Musen M.A. Ontology versioning in an ontology management framework. *Intelligent Systems, IEEE*, 19(4), pp. 6-13, 2004.
- [2] McGuinness, D. L., Fikes, R., Rice, J., & Wilder, S. An Environment for Merging and Testing Large Ontologies. In: *Proc. of the 7th Int. Conf. on Principles of Knowledge Representation and Reasoning, KR 2000*, pp. 483–493, 2000.
- [3] Hatala, M., Gasevic, D., Siadaty, M., Jovanovic, J., & Torniai, C. Ontology extraction tools: An empirical study with educators. *Learning Technologies, IEEE Transactions on*, 5(3), pp. 275-289, 2012.