



Harmonizing Software Standards with a Semantic Model

Verhelst, E. C. H., Lekkerkerk, H. J., Bulens, J. D., & Janssen, S. J. C.

This is a "Post-Print" accepted manuscript, which has been published in "None"

This version is distributed under a non-commercial no derivatives Creative Commons



([CC-BY-NC-ND](https://creativecommons.org/licenses/by-nc-nd/4.0/)) user license, which permits use, distribution, and reproduction in any medium, provided the original work is properly cited and not used for commercial purposes. Further, the restriction applies that if you remix, transform, or build upon the material, you may not distribute the modified material.

Please cite this publication as follows:

Verhelst, E. C. H., Lekkerkerk, H. J., Bulens, J. D., & Janssen, S. J. C. (2011). Harmonizing Software Standards with a Semantic Model. In E. García-Barriocanal, Z. Cebeci, M. C. Okur, & A. Öztürk (Eds.), *Metadata and Semantic Research*, 5th International Conference, MTSR 2011, Izmir, Turkey, October 12-14, 2011 (Vol. 240, pp. 170-177). Heidelberg: Springer. <https://doi.org/10.1007/978-3-642-24731-6>

Harmonizing Software Standards with a Semantic Model

Elisabeth Verhelst¹, Huibert-Jan Lekkerkerk², Jandirk Bulens¹, and Sander Janssen¹

¹ Alterra, Environmental Sciences Group
Wageningen University and Research Centre
Droevendaalsesteeg 3, 6708 PB Wageningen, The Netherlands
² Informatiehuis Water
Stationsplein 89, 3818 LE Amersfoort, The Netherlands

Abstract. The application of standards in the software development process supports interoperability between systems. Maintenance of standards must be guaranteed on the organisational and technical level. The use of semantic technologies can contribute to the standard maintenance process by providing a harmonizing bridge between standards of different knowledge domains and languages and by providing a single point of administration for standard domain concepts. This paper describes a case study of the creation of a semantic layer between software standards for water management systems in The Netherlands.

Keywords: interoperability, standard maintenance, semantic, ontology

1 Introduction

Interoperability between software systems is supported by the adoption of standards in the software development process. Standards are available on different levels of abstraction: national and international, domain specific and software development methodology specific. It is essential to keep the standards aligned and coherent to avoid interoperability problems. Standard harmonisation activities can be observed at the organisational and technical level. On the organisational level this is arranged via organisational bodies such as advisory groups and forums of standard organisations. On the technical level a variety of instruments are necessary. We mention here syntactical and semantic measurements. This paper introduces a harmonisation method for standards on the semantic level. It is based on the creation of a semantic mediating layer between a set of existing software standards.

Semantic mediation has been proposed as a method to achieve integration in multi-disciplinary and multi-scale environments [1];[2]. Through semantic mediation an agreed conceptual schema can be built between parties for linking their systems [2]. The agreed conceptual schema serves as an additional knowledge layer. Semantic mediation is a crucial challenge for any integrated project

[3]; [4]; [5], as it provides consistent and transparent building blocks in definitions and terms required for the methodological and technical linking of systems, data sources and indicators. Usually semantic mediation is based on an ontology, which is a specification of a conceptualization for a system [6]. An ontology consists of concepts and their relationships [7]. This paper focuses on the use of ontologies for semantic integration of standards.

2 Problem Statement

Standards are heavily interconnected. Progress in technological developments and changing requirements determine a continuous harmonisation necessity. It is a challenge to determine whether and how a change in a revised standard affects other standards. Examples of general proposed standard revision areas are: scope, normative reference, terms and provisions [8]. An analysis of standard interdependencies enables an in-depth understanding, necessary for standard maintenance. In addition, version control and change management procedures look after a reproducible, documented and structural standard maintenance process. On the syntactic level this means that standards have to remain consistent in the ability of exchanging information. Semantic harmonisation regulates the interconnection of standards between knowledge domains and languages. Ambiguity and redundancy in terms and definitions must be avoided. As an example of an organisational semantic harmonisation activity we mention here the designated maintenance agency at ISO, responsible for guarding the consistency of standard meta-metadata, such as country names, language names, character sets³. Our research aims to contribute to an automated manner of harmonisation of the terms and concepts within standards.

3 Software Development Methodology and Semantics

The importance of semantics in software development has been recognised for several stages in the development process. Research has been conducted in order to integrate semantics in the requirements specification phase [9]. Use of semantics has been proposed for process modeling [10]. In 2009 the organisation for development of enterprise integration standards Object Management Group (OMG) published the Ontology Definition Metamodel (ODM) [11]. The ODM aims to regulate 'the formal grounding for representation, management, interoperability, and application of business semantics...' by offering amongst others '...Options in the level of expressivity, complexity, and form available for designing and implementing conceptual models, ranging from familiar UML and ER methodologies to formal ontologies...' [11], p. 1). The ODM describes the relationship between the Model Driven Architecture (MDA) components such as UML and Common Logic (CL) and knowledge representation languages like OWL. Many research activities have contributed to documenting methods of

³ http://www.iso.org/iso/maintenance_agencies

integrating semantic technologies with MDA[12]; [13]. In [14] the differences between OWL and UML have been unraveled. In our research we have concentrated on the meaning of terms and concepts used in the standards. In a case study we developed a semantic concept harmonisation layer between the Aquo standards in order to contribute to the interoperability of the standards on the semantic level.

4 Case Study

The Netherlands is a country that has a significant portion of its total area below sea level. The country has also many water bodies. Proper water management is therefore essential. For this purpose the Dutch water boards and the national government share several information systems. The Dutch 'Informatiehuis Water' (IHW) is responsible for setting and maintaining the standards for the development of these systems. This guarantees that water management organizations create systems that can interconnect and exchange information effectively. IHW maintains many standards, identified by a common name 'Aquo standard'. Every standard has its own purpose. The standards are related on the international level (ISO), national level (NEN3610) and via domain specific information models (IM) (IMRO, IMWA etcetera) (Fig. 1) The Aquo standard IMWA is part of the NEN3610 standard.

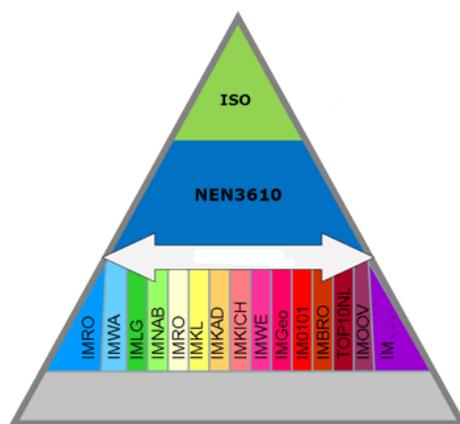


Fig. 1. Dutch standard NEN3610 and its dependencies

Details of the separate Aquo standards are given in Table 1. Though different in application, syntax format and maintenance software, all parts of the Aquo standard refer to the same domain concept origin. In Table 1 this is elaborated for the concept '*sluice*'. The objective of the research was to provide for a semantic

layer that contains the concepts (e.g. 'sluice') and defines the relations of those concepts within the group of standards. At the same time a single point of concept maintenance was aimed for.

Table 1. Details of the separate standards in the Aquo Standard. The example column gives an example of the notation of the concept 'sluice' in the standard.

Name	Application	Syntax Format	Example
Aquo-lex	Thesaurus	XML	<pre><Code>KSL, 1009</Code> <Term>sluice</Term> <Definition>'An artificial , movable barrier that can open or close the connection between two waters '</Definition></pre>
UM Aquo	Exchange of information	UML	FeatureType with Attribute TypeOfWatermanagementBody (retrieved from domain table list)
LM Aquo	Logical model	ER	Entity = sluice Attribute = type of sluice; capacity; location
IMWA	Exchange of geo information	UML	Feature Type with Attribute geometryPoint
Domain tables	Keyword lists	XSD	<pre><xs:enumeration value=" Watermanagement Body; sluice" /></pre>

5 Results

The general approach was to first investigate which concepts in the standards were overlapping, how they were related and next to design a common semantic standard, or ontology. The central starting point was Aquo-lex, since it is by its origin a thesaurus containing concept definitions and relations. The Simple Knowledge Organization System (SKOS) [15] model was used to semantically disclose the concepts into a new version of Aquo-lex. SKOS is a powerful, well-ordered KOS model that can store semantic relations such as

broader and narrower, as well as language specifications, definitions and synonyms. KOSses from different domains (schemes) can be connected together via the `skos:ConceptScheme` class. In order to link the concepts from the SKOS Aquo-lex to the other Aquo standards some extra OWL classes and properties needed to be defined. These properties explicitly describe the linking relation in a semantic way, e.g. `hasConcept` is the link of the ER diagram to the concept definition in SKOS (Fig. 2).

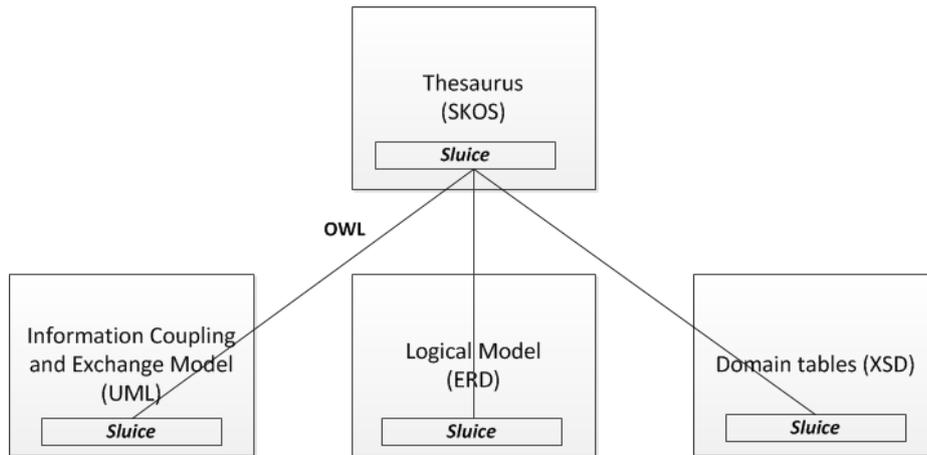


Fig. 2. The linked concepts in the separate standards, with the lines between the Sluice boxes identifying the semantic relation `hasConcept`

Once this SKOS-OWL data model was designed it had to be loaded with the actual values of the Aquo-lex standard (the SKOS part) and the values of the related linked standards (the OWL part). The conversion of Aquo-lex XML to SKOS was executed by means of a data conversion product⁴. In addition XSLT was used to reformat the conversion result to proper SKOS syntax. The conversion of ER and UML to OWL was done by hand on a small set of instances, as this was the best way to ensure proper match between concepts for the relatively small number of concepts. The values were given unique ids in order to make them independent of language and character coding limitations. For example: concepts were coded as `CONxxxx` and entities from the ER diagrams with `ENTxxx` (xxx referring to a number sequence). The result of the construction of the semantic layer was an OWL file that contained individuals of SKOS concepts originating from Aquo-lex and their relations with the rest of the Aquo standards. Below is listed a part of the OWL file, in which the concept with id `CON0001` is related via the `hasConcept` property to the EntityClass with

⁴ FME from SAFE software (<http://www.safe.com>)

id ENT001. The named value 'SLUICE' can be observed via the `rdfs:label` property.

```
<owl:Thing rdf:ID="ENT001">
<rdf:type rdf:resource="#EntityClass"/>
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>SLUICE</rdfs:label>
<catalogus:code
rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>KVP</catalogus:code>
<catalogus:hasConcept rdf:resource="#CON001"/>
</owl:Thing>
```

The creation of this semantic layer has resulted in the conversion of the Aquo-lex standard from XML to SKOS. It can eventually lead to the elimination of the stored concepts in the other Aquo standards. In return these other standards are to be provided with linked concepts that are administered via the SKOS-OWL reository. The contents of the SKOS-OWL file was demonstrated via a graphical user interface (Fig. 3) that showed the concept definitions in the new SKOS Aquo-lex (central in the picture) together with the information of the related standards (surrounding frames).



Fig. 3. A graphical user interface on top of the semantic layer (in Dutch)

6 Conclusion and discussion

We have created a semantic model for harmonising concept definitions between software standards with SKOS-OWL. By itself SKOS does not provide sufficient

properties to define concept relations beyond the thesaurus level. However the openness of the OWL syntax allows for addition of user-defined properties. We used this possibility in our SKOS-OWL design to add OWL properties that link concepts between standards. The developed semantic model is a first step in the harmonization of terms and concepts in the Aquo standards. While the GUI helps to make it accessible to a wider audience, further steps in the development of the Aquo standards are on integrating the syntax content of the different standards more tightly (see column 4 of Table 1). This would require clear rules for conversion between different programming paradigms, such as OWL-UML, OWL-ER (see, for example, [16]) and OWL-XSD translations and vice versa. In addition procedures need to be set up to ensure the consistency of such a set of linked standards and ontologies, as changes in one standard can affect other standards. Tools for easy editing need also be available, as ontology editors can be difficult to use for those not familiar with such tools or the concept of ontologies.

A semantic concept model can be used for two purposes. The first is to simplify standard concept definition and maintenance by means of a single concept administration point. The second purpose is to align standard concepts from different domains using the `skos:ConceptScheme` class. We suggest the development of a separate ontology specification per domain standard (similar to the result of our research), and then to link the separate ontologies into an umbrella ontology that combines all the concepts in the subsequent standards. The umbrella ontology can be used as a semantic bridge between the domain specific standards as depicted as the horizontal arrow in Fig. 1. The two purposes described contribute to the semantic interoperability of standards.

Acknowledgments We gratefully acknowledge the financial support of Informatiehuis Water for carrying out this research. The authors would like to thank all colleagues who contributed to this study. We are grateful to the standard specialists Wilbert Vos, Hinne Reitsma and Willemien Joosse of IHW for their advice and the introduction to the Aquo standards. We would like to thank Arend Ligtenberg for his valuable feedback during preliminary investigations. We also thank our Software Engineers Steven Hoek, Inge la Riviere, Jappe Franke and Matthijs Danes for building the software implementation.

References

1. Rizzoli, A. E., M. Donatelli, et al. : Semantic links in integrated modelling frameworks. *Mathematics and Computers in Simulation* 78(2-3): 412-423. (2008)
2. Villa, F., I. N. Athanasiadis, et al. : Modelling with knowledge: A review of emerging semantic approaches to environmental modelling. *Environmental Modelling and Software* 24(5): 577-587. (2009)
3. Jakobsen, C. H. and W. J. McLaughlin : Communication in Ecosystem Management: A Case Study of Cross-Disciplinary Integration in the Assessment Phase of the Interior Columbia Basin Ecosystem Management Project. *Environmental Management* 33(5): 591-605. (2004)

4. Bracken, L. J. and E. A. Oughton : 'What do you mean?' The importance of language in developing interdisciplinary research. *Transactions of the Institute of British Geographers* 31(3): 371-382. (2006)
5. Scholten, H. : "Better Modelling Practice: an ontological perspective on multidisciplinary, model based problem solving". Wageningen, Wageningen University. PhD: 313. (2008)
6. Gruber, T. R. : A translation approach to portable ontology specifications. *Knowledge Acquisition* 5: 199-220. (1993)
7. Antoniou, G. and F. van Harmelen: A semantic web primer. Cambridge, Massachusetts; London, England, The MIT Press. (2004)
8. Coetzee, S. : Results from a normative dependency analysis of geographic information standards. *Computer Standards and Interfaces* 33(5): 485-493. (2011)
9. Kaiya, H.; Saeki, M.; , "Ontology based requirements analysis: lightweight semantic processing approach," *Quality Software*, 2005. (QSIC 2005). Fifth International Conference on , vol., no., pp. 223- 230, 19-20 Sept. 2005 <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1579139&isnumber=33358>
10. Rodriguez-Garca, D., Barriocanal, E.G., Alonso, S.S., Nuzzi, C.R.: Defining Software Process Model Constraints with Rules Using OWL and SWRL. *International Journal of Software Engineering and Knowledge Engineering*(2010) 533-548
11. OMG: *Ontology Definition Metamodel Version 1.0*. (2009)
12. Gašević, D., V. Devedžic, et al. : *Model Driven Architecture and Ontology Development*. Berlin, Heidelberg, Springer-Verlag Berlin Heidelberg. (2006)
13. Alvarez, M. A., B. C. P. G-Bustelo, et al. : *Bridging together Semantic Web and Model-Driven Engineering. Distributed Computing and Artificial Intelligence*. A. P. D. DeCarvalho, S. RidriguezGonzalez, J. F. D. Santana and J. M. C. Rodriguez. Berlin, Springer-Verlag Berlin. 79: 601-604 (2010)
14. L. Hart, P. E., B. Colomb, K. Raymond, S. Taraporewalla, D. Chang, Y. Ye, E. Kendall, M. Dutra: "OWL full and UML 2.0 compared." (2004)
15. Isaac, A. and E. Summers: *SKOS Simple Knowledge Organization System Primer*, W3C (2009)
16. Athanasiadis, I. N., F. Villa and A. E. Rizzoli : *Enabling knowledge-based software engineering through semantic-object-relational mappings*. 3rd International Workshop on Semantic Web Enabled Software Engineering, 4th European Semantic Web Conference. Innsbruck, Austria. (2007)