A Semantic Web-based Approach for Representing and Reasoning with Vocabulary for Computer-based Standards Processing

Salih Demir, James H. Garrett, Jr., Burcu Akinci and Omer Akin
Carnegie Mellon University, Pittsburgh, PA, USA

Mark Palmer
National Institute of Standards and Technology, Gaithersburg, MD, USA

Abstract

Although web technologies and information systems have matured at an increasing rate, the users of engineering codes and standards have yet to benefit to the fullest from such advanced information processing and reasoning capabilities. This is partially due to the difficulty of finding appropriate syntaxes to represent standards that are easily understood and able to be processed by a computer and that are testable and maintainable by standards writing organizations. Most of these codes and standards are only available as text-only documents (either in printed form or as pdf files) that do not formally represent the underlying semantics of the text. Hence, it is difficult to conduct any formal computer-based reasoning about the content of these documents in this form. There are some recent developments, such as SMARTcodes, that target these issues and represent both the vocabulary and the code logic. However, since the represented vocabulary terms might not always match the terms used in the building model representations, there is a need for generating mappings between the vocabulary and the standardized building models. In this paper, we discuss how the Semantic Web technologies can help in representing and mapping vocabulary terms to building information models. We also discuss how well the developed approach performs on a portion of ASHRAE 90.1 Energy Standards for Buildings.

Keywords: design, standard, conformance checking, mapping, semantic web

1 Introduction

The formal representation of codes and standards has been an area of study by a large number of researchers (see the overview paper by Fenves et al., 1995). A variety of different approaches to representing formally the logic of a building code or standard were explored in previous research. Most of this research faced the challenges of not having a standardized building model representation and a comprehensive and reusable vocabulary in computable form. Building information modeling and common vocabulary development efforts, such as IFD (Bell et al., 2008) and e-Cognos (Lima et al., 2004), provide new mechanisms and much needed implementations and momentum for standards processing research. These efforts are attempting to fill the aforementioned gaps. However, some of the most significant challenges in this area today are: 1) the development of a common vocabulary that can be mapped to the standardized building model representations, and 2) creating syntaxes for the standard and vocabulary representations that are easily testable and maintainable by standard writing organizations.
There are some recent developments, such as the SMARTcodes initiative (Nisbet et al., 2008) by the International Code Commission (ICC), that can be used to assist in the creation of electronic versions (in the form of a rule set) of building energy standards. The SMARTcodes initiative has been able to address the requirement of simplicity of standard representation syntax even though the mapping is currently handled by embedding its logic within the program code (i.e. it is hard-coded and not made explicit as part of the vocabulary itself) in the current version. In the case of new vocabulary terms generated when a standard is updated or new standards are created, a standard organization has to hire a software engineer to handle the mapping of the new vocabulary terms.

There have been significant efforts to address some requirements of the standard representations by introducing a Web Ontology Language (OWL) notation for the IFC (IAI, 2000) and then for the representation of the standards (Beetz et al., 2009; Yurchyshyna et al., 2008). Using OWL in this way allows Semantic Web related products, such as SPARQL (Prud'Hommeaux and Seaborne, 2008), to handle the mapping problem. Although introducing OWL notation for the IFCs and the representation of the standards provided explicit mappings, the approaches as described in the literature do not appear to offer a simple syntax for the representation of the standards. In addition, the vocabulary and its mapping become embedded in the standard representation which requires standard writing organizations to redefine their vocabulary terms within the many standards that they are modeling. There is a need for an explicit and simple vocabulary representation that also handles the mapping to building models in a simple and explicit manner. Such an approach will allow the standard modelers to reuse the vocabulary in several different computer-processable models of standards for which the vocabulary is relevant. Besides, the standard writing organizations will be able to maintain the vocabulary more easily and without if it is explicitly represented once and used many times.

2 Requirements

In order to increase the maintainability and reusability of software components and decrease the overall programming effort in the lifecycle of software, some principles coming from experience are reused many times. The General Responsibility Assignment Software Patterns (GRASP) is a collection of principles that guides software designers in assuring good practice for their designs (Larman, 2005). The principle of low coupling, which keeps the dependencies among software components as minimal as possible, and the principle of high cohesion, which requires that the components have related and focused responsibilities, are utilized to create maintainable and reusable software modules resistant to changes in other components (also referred as protected variation). These principles translate into representing the provisions as independent components of a standard, which is also referred to as isomorphic modeling of standards by Kiliccote and Garrett (1998). By treating provisions as independent components, the standard representations become not only easy to comprehend and maintain, but also as close in form to the corresponding text as possible. Larman calls these types of representations as having a low representational gap (Larman, 2005). Standard models with a high representational gap are hard to validate and less reusable.

According to the GRASP principles, another good standard modeling practice would be representing the standard vocabulary explicitly rather than embedding it in the standard models. Embedding the vocabulary will create many repetitions in the standard models even though there is an opportunity to reuse the vocabulary in many provisions from many standards. This is similar to the GRASP principle of polymorphism, where designs have polymorphic classes consisting of core methods, which are applicable to all subclasses, and polymorphic methods, where designers leave the reasoning in the methods to the individual subclasses. There is a need to have polymorphic standard representations, where the vocabulary is one of the core components and it is reused among several standard provisions from several standards in order to avoid repetitions and high impacts from the possible changes in the vocabulary. On the other hand, the process of reasoning with the vocabulary would be left to the standard representations as in the case of abstract methods in polymorphic classes.
There is also a need to apply the GRASP pattern *indirection*, which is used to assign the responsibility of mediation between classes to an intermediate class and ensure low coupling between these classes. By applying this principle, a change happening in one of the classes would not affect the other class and can be handled using a modification to the intermediate class. This idea is applicable to the mapping between the vocabulary and standardized building model representations that will make the vocabulary independent from any building model representation and allow the standard representation to work with any building model representation.

3 Current Approaches

The requirements of a standard vocabulary were presented in the previous section. The SMARTcodes approach is able to address the requirement of understandability and maintainability of the standard representations, since it is an isomorphic approach and it maintains the vocabulary explicitly. However, the mapping is left to the software that processes the vocabulary and standard and there is no explicit representation of the mapping in the current version. Yurchyshyna et al. 2008, on the other hand, did not discuss a general standards processing architecture that maintains a vocabulary representation. However, they did discuss a Semantic Web based approach for handling the mapping by SPARQL queries that is able to represent the mapping explicitly.

There are many other approaches we do not cover here in detail that can be classified as, procedural, rule-based, object-oriented, context-oriented, logical and hybrid approaches (Rasdorf and Wang, 1988; Jain et al., 1989; Yabuki and Law, 1993; Kiliccote, 1994; Liebich et al., 2002). All these approaches so far helped the current hybrid standards processing approaches in many ways. However, there is no approach, among those listed that we know of, that is able to address all of the requirements we identified. In this paper, we only focus on the SMARTcodes approach and that used by Yurchyshyna et al. 2008 since we are planning to build upon these approaches in order to address the requirements identified in Section 2.

4 Why the Semantic Web?

There are several systems that a standard processing system may need to interact with, such as standard representations, building model representations, online product catalogs and simulation/analysis software. Therefore, interoperability of a standard representation is an important criterion. Hypothetically, interoperability can be achieved by creating a common vocabulary using XML (www.w3.org/XML) for all the representations the participating systems utilize and having all relevant parties agreeing on the formal semantics of the vocabulary and the syntactical structure created. However, this is not a realistic approach, since different systems have different interpretations of specific objects and need different levels of detail in their representation. However, it is possible that all relevant parties can create their current hybrid standards processing approaches in many ways. However, there is no approach, among those listed that we know of, that is able to address all of the requirements we identified. In this paper, we only focus on the SMARTcodes approach and that used by Yurchyshyna et al. 2008 since we are planning to build upon these approaches in order to address the requirements identified in Section 2.
www.w3.org/RDF), introduced a graph-based foundation for the Semantic Web to define formal semantics using subject, predicate and object triples. This is powerful since the parties can now define relationships between concepts in a machine-interpretable manner and navigate the graph by tracking the relationships. Ontology-level constructs, which were not introduced by RDF/RDFS, were introduced in OWL, such as equivalence/inequivalence, relationship characteristics, restrictions and set operations. The query and rule languages further increase the capabilities of the Semantic Web by allowing users to extract sub-graphs from RDF/RDFS graphs and define new rules.

The Semantic Web is relevant for the creation of standards processing frameworks since it would enable these frameworks to interoperate and exchange data with other relevant systems by using a standard ontology framework that also provides query and rule languages. In order to do this, the representations needed for an automated standard checking system need to be defined in OWL. We present a possible OWL notation for the vocabulary and mapping representations in the next section.

5 Research Method

5.1 Proposed Architecture

Figure 2 illustrates one possible architecture for an automated standard checking system that is able to explicitly represent and map a vocabulary to the building model representations. The system consists of a standard model, a building model, the definition of the vocabulary, a mapper, a model view generator (MVG) and model checking software (MCS). Since the approach presented in this paper focuses on the vocabulary representation and reasoning for standards processing, we do not provide a detailed description of the MVG and MCS, but rather state the role of these components in the proposed system architecture. The standard, the vocabulary used in that standard and the building model to which the standard is applied are data models and the rest are dynamic components. The model view generator (MVG) makes inferences from the building model facts by interacting with
related simulation/analysis engines, which are sometimes needed to provide a value for a given terms stated in the vocabulary, such as terms requiring an energy simulation or terms that require a view generation, and hence cannot be directly mapped to the building model. The mapper consists of multiple layers that have mapping information between vocabulary terms and potential building model representations, e.g., IFC, gbXML. These layers consist of SPARQL queries that are used to map vocabulary terms to the specified building model representation in the layer (explained in detail in Section 5.3). The model checking software (MCS) determines applicable requirements for the applicable building components and checks for conformance.

5.2 Vocabulary Representation

In order to begin to determine what is needed for representing a vocabulary used in a standard, we evaluated the American Heating Ventilating and Air Conditioning Engineers (ASHRAE) Energy Standard for Buildings Except Low-Rise Residential Buildings (90.1 2007), in addition to searching and identifying representations and any existing vocabularies used in the thermal codes domain. What we observed as critical to code conformance were generalization-specialization (taxonomic) hierarchies, holonym-meronym (aggregation) hierarchies, and synonymical relationships. As these hierarchies and equivalence classes get richer, the standard and mapping representations get simpler.

It is possible to represent taxonomical and synonymical relationships using two built-in OWL relationships: rdfs:subClassOf and owl:equivalentClass. However, OWL does not have built-in constructs for representing aggregations or term attributes, which are important for product modeling ontologies, as well as standard representations, since they specify requirements for products. OWL allows for the creation of user-defined relationships even though the user is expected to handle most of the reasoning. For example, the OWL reasoners will automatically handle inheritance for taxonomical relationships, but they will not infer aggregation-related rules for holonym-meronym hierarchies unless these are specified with additional rules using a Semantic Web rule language. Attributes need to have names, data types, units and possibly some limit values. OWL is able to represent attributes using triples, but all of the outgoing edges of a subject will not indicate a relationship that assigns an attribute to the subject, (e.g. svc:Term2 rdfs:subClassOf svc:Term1 is not assigning an attribute to svc:Term2, as shown in Figure 3). It would be hard to answer queries like “What are the attributes of term ‘Wall’?” without representing the attribute concept by defining svc:Attribute class and svp:hasAttribute relationship as shown in Figure 3.

![Figure 3. A portion of the vocabulary implemented in OWL](image-url)

We used the ASHRAE 90.1 glossary as a test-bed and prepared an OWL model of a portion of it that addresses the vocabulary representation requirements we identified in Section 2 and can be used in standard representations. Figure 3 illustrates a simplified version of the important triples in the OWL representation of the vocabulary we created. The illustration does not show class and
relationship definitions, representation of term definitions using rdfs:comment and representation of synonymical relationships using owl:equivalentClass due to space limitations. There are also other triples we are not able to list here. The full names of the classes cannot be used in RDF if they contain spaces and other characters that are not allowed as RDF terms, as in the case of “Exterior Wall”. These types of terms can be only leaf nodes in the graph, called literals. Instead, it is common to use identifiers, such as svc:Term, and assign the actual names using svp:hasName relationship.

5.3 Mapping of the Vocabulary

Model-to-model mapping in general is a challenging problem, which has been studied by many researchers. The challenge comes from the fact that there are 20 different categories of mappings (Katranuschkov, 2001). However, according to Katranuschkov (2003), these 29 classes of mappings can be grouped into three main categories: unconditional class-level mappings; conditional instance-level mappings; and attribute-level mappings. Conditional instance-level mappings involve a subset of the instances of the source and target classes by applying a constraint, unlike class-level mappings and attribute-level mappings that specify how to map the attributes of classes.

It is possible to handle these mappings by querying the target model using an advanced query language. SPARQL is a W3C recommended query language that can be used to affect the mappings so long as we have an OWL version of the vocabulary definitions and the building model representations. Since OWL notation of IFCs has already been introduced (Beetz et al., 2009), we focus on ifcOWL to test our approach. Extraction of sub-graphs from ifcOWL building models has already been presented and tested by several researchers (Beetz et al., 2009; Yurchyshyna et al., 2008). We also adopt the same approach by handling the mapping using SPARQL. Figure 4 is an example of a SPARQL query that extracts the instances of “Exterior Wall” (svc:Term2 in Figure 3) of the standard vocabulary from the ifcOWL representation of building models. This query can be translated as “Select IfcWall instances where the value of ‘IsExternal’ attribute is indicated as ‘True’”, which is a conditional instance-level mapping according to Katranushkov’s mapping classification.

```
SELECT ?wall
WHERE {
  ?wall a ifcsharedbldgelements:IfcWall.
  ?prop a ifckernel:IfcPropertySet.
  ?sProp a ifcpropertyresource:IfcPropertySingleValue.
  ?sProp ifcpropertyresource:Name "IsExternal".
  ?sProp ifcpropertyresource:NominalValue "True".
}
```

Figure 4. An example SPARQL query for extracting “Exterior Wall” instances from ifcOWL

Most of the lines in queries for ifcOWL are used to drill down to the leaf nodes of the IFC model due to the fact that IFC is a large model consisting of deep hierarchies. Future research for us might be to simplify mapping of standard vocabulary terms to building model representations by defining an intermediate representation that removes the unneeded parts for standard modeling and is easily understandable and transferrable to SPARQL. In this intermediate representation, the terms in the vocabulary themselves can be used to avoid redundancies, e.g., svc:Term1 (the identifier of “Wall” in Figure 3) can be referenced in the query for mapping svc:Term2 (the identifier of “Exterior Wall” in Figure 3) instead of redefining svc:Term1 in the svc:Term2 query. For unconditional class-level mappings, only a list of the names of the corresponding classes would be sufficient, e.g., “IfcWall” would be sufficient to map “svc:Term1” to IfcWall. Even though the simplicity will cause less expressive queries in general, it will satisfy the requirements of standard modeling.
Closure

In this paper, the requirements of the vocabulary for computer-based standards processing were presented. There is a need for an explicit and simple vocabulary representation that also handles the mapping in a simple and explicit way. A vocabulary and mapping representation that is simple and explicit are proposed in this paper. This approach will eliminate the redundancies in the standard representations and allow standard writing organizations to reuse the vocabulary in several computer-processable versions of their standards. The standard writing organizations will also be able to maintain the vocabulary with less difficulty. We are also planning to implement an automated standard conformance checking system whose architecture is illustrated in Figure 4 and explained in Section 5.1.

Acknowledgements

The project is funded by a grant from the National Institute of Standards and Technology (NIST), Grant # 60NANB8D8159 and is gratefully acknowledged. Any opinions, findings, conclusions or recommendations presented in this paper are those of the authors and do not necessarily reflect the view of the National Institute of Standards and Technology.

References


