Modeling Vocabularies in the Architectural Domain

Katja Niemann, Martin Wolpers
Fraunhofer Institute for Applied Information Technology
Schloss Birlinghoven, 53754 Sankt Augustin, Germany
{niemann, wolpers}@fit.fraunhofer.de

Abstract

The aim of the MACE project is to overcome the fragmentation of digital learning resources in architectural education. This requires the human-controlled enrichment of learning resources, i.e. the tagging of learning objects with controlled architectural vocabularies. In this paper, we briefly describe how the architectural vocabulary used for manual enrichment is modelled. Furthermore, we develop the requirements for setup and maintenance of these vocabularies. Finally, we briefly compare existing maintenance tools and reason why and describe how we use the Protégé tool to maintain the MACE classification vocabulary.

1. Introduction

The use of digital learning resources in architectural education is rather fragmented today. Learning resource repositories exist next to each other without any direct relationship thereby hampering access significantly. By enabling architecture students to access learning resources in a uniform way, such fragmentation is overcome. Furthermore, a larger community is created that contributes as well as draws on the excellent digital architectural learning resources provided today. In order to use learning resource repositories in a seamless way, we rely on the usage of metadata descriptions of the learning resources. Learning resources are made available uniformly through a central metadata store that harvests the learning resource metadata from each connected repository [12]. Different repositories provide different amounts of metadata, human-controlled metadata enrichment processes are employed to create a coherent and concise metadata base that describes the available learning resources concisely.

The work described here is carried out within the MACE project (Metadata for Architectural Contents in Europe) [2]. The aim of MACE, a European project co-funded by the European Commission, is to simplify the sharing of digital learning materials across architecture schools in Europe. Within MACE, already existing repositories of architectural digital learning resources are linked, thereby allowing easier finding and access to these learning materials. The infrastructure of MACE uses metadata about the learning resources extensively. The federation of repositories in MACE is based on the MACE application profile, a metadata schema extending LOM [1] that is used to unify the metadata representations from the participating learning resource providers.

The metadata is harvested from the providers into the central MACE metadata repository. Employing the MACE specialised enrichment tools, the metadata is automatically and manually enriched by architects to correct existing and to provide new metadata. The variety of metadata added to each learning resource ranges from technical and educational descriptions to rights management and classifications.

Here, we concentrate on the manual enrichment of learning resources with classification terms from architectural vocabularies. A concise vocabulary serves as basis for the communication within MACE, between users, experts and MACE systems, thus following the principles outlined in [15]. The vocabulary is built from a number of architectural online and offline classifications like the Getty architectural thesaurus [3] and the Ci/SfB construction index [6]. As these online accessible indices are mixed and extended with own vocabulary in MACE, we have decided to setup our own MACE vocabulary store. This store provides access to and maintenance of more than 2.000 MACE vocabulary terms organized in hierarchies with at most 5 levels.

The vocabulary was set up manually. Several problems surfaced including the creation, manipulation and deletion of terms and structure through a group of experts. Consequently, manual role back and back up procedures were put in place. The vocabulary and its structure had to be translated into an electronic format understood by the web-based enrichment tool.

978-1-4244-2917-2/08/$25.00 ©2008 IEEE

Quite expectedly, the task of maintaining and using the vocabulary became a too complex and too error-prone task to be done manually. It was quickly decided to use computer-based tools to support maintenance tasks. This paper outlines the experiences made in maintaining the MACE vocabulary using a semi-automatic approach with architecture experts not familiar to the task of maintaining a classification. In chapter 2, we describe an excerpt of the classification maintenance software we had a look at, our requirements and why we chose Protégé. In chapter 3 we describe the MACE vocabulary and its representation. Chapter 4 shows how the vocabulary is maintained with Protégé and chapter 5 concludes with a summary of our experience so far.

2. Classification maintenance software

The use of classification maintenance software must simplify the maintenance of the vocabulary by providing methods for creating, editing, storing, querying and deleting vocabulary entries. These are common requirements for classification management software. Consequently, there are many free systems which meet these criteria.

However, the MACE vocabulary service requires additional features. The application should support the import and export of the vocabulary using a common format in order to be able to intervene in the maintenance by using manually created vocabularies. Furthermore, the software should have a database backend to provide features like transaction rollback, backup and fast access to large vocabularies to be able to track and possibly reverse changes that are made on the vocabulary.

In addition, in MACE vocabulary maintenance is a community task and carried out by a number of expert architects. Consequently, it is essential to ensure that everyone works on the same version of the vocabulary. Therefore, the service needs to provide a web frontend which supports simultaneous collaborative work on the classification. User management and access rights need to be put in place to protect the emerging classification against possibly wrong changes by non-accredited users. The vocabulary service is used by a number of MACE front end services (e.g. the web-based MACE content enrichment tool) to allow the manual classification of learning resources. As the MACE infrastructure is realized as a service oriented architecture, the vocabulary has to be accessible via a web service API to provide services like search and translation.

As we decided to use a classification management software we consequently had a look at different tools to see which tool satisfy our needs best. Here we compare a few examples.

Apollo CH [9] is developed at the Gerstner Laboratory of the Czech Technical University, Prague, Czech Republici. The internal model is built as a frame system according to the knowledge model of the Open Knowledge Base Connectivity protocol (OKBC) [5]. In this model, an ontology consists of a set of classes which are organized in a subsumption hierarchy and represent the main concepts of a domain, a set of slots associated to classes to describe their properties and relationships and a set of instances of those classes holding specific values for their attributes [5]. The default storage format is XML, but Apollo can be adapted to support different storage formats via I/O plug-ins. If multiple users work on an ontology, a repository has to be used and the locally stored ontologies have to be synchronized if someone changes something. Additionally, Apollo does a full consistency check while editing the ontology. However, Apollo does not offer a web service which can be used from the MACE back-end services.

Ontolingua [10] provides an environment to browse, create, edit, modify and use ontologies. Ontolingua is only available over the World Wide Web and is accessible via a standard web browser. The ontologies are stored on the server, but it is possible to export them in a LISP-like code or as RTF file. Furthermore, Ontolingua supports collaborative work on an ontology. But as Apollo, Ontolingua does not offer a web service interface.

Protégé [5] is a free, open source ontology editor, developed at Stanford Medical Informatics, Stanford University. It is written in JAVA and its plug-and-play environment makes it an easy to use application to build customized tools for ontology construction and management. The Protégé Frames knowledge model is also compatible with the OKBC [5]. Consequently, the knowledge model of Protégé Frames forms a good basis for terminology systems [11]. But first of all, Protégé seemed to support all our demands related to the collaborative work possibilities and to a web service. Furthermore, Protégé is continuously enhanced and supported from a large community.

Hence our requirements and the need of a cost-free tool resulted in the decision to use Protégé.

3. MACE vocabulary

Here, we introduce how vocabulary values are described in MACE and how we model the vocabulary in Protégé for maintenance and reference.

First, a brief example describes the nature of the vocabulary values that we are dealing with.

Figure 1: Image of the UFA Cinema Building in Dresden, Germany

The above example learning resource shows an image of a multi-cinema building [7]. Using the MACE classification vocabulary, the architects classify the building shown in the learning resource with structured terms like {functional typology = recreational facilities.cinema} and {material = glass, metal.steel}, the full classification is available at [7].

The vocabulary consists of several taxonomies, called groups, e.g. “Conceptual Design” or “Technical Design”, which are combined through a root element called “LOM Category 9 Classification”. The groups are split into categories, e.g. “Technical Design” is split into the categories “Material”, “Construction Form”, “Structural profile” and several others. These categories contain the terms which for the most part can be used for classification purposes (see Fig. 2 for clarification).

The MACE classification vocabulary consists of 2.777 base terms which have children with a depth of up to 5. Therefore, each item of the MACE classification vocabulary, e.g. “cinema”, must be an instance of the ontology used to model the vocabulary. Basing our model on the Neuchatel Terminology Model [13], we nevertheless use a highly simplified version to satisfy the MACE vocabulary needs.

The MACE vocabulary consists of terms that can have terms as children and a number of additional attributes. The root (and only) class of our model of the vocabulary represents the “term” concept of vocabularies. No further classes are defined as this root class suffices for the purpose of the MACE vocabulary. The root class is called “Base Element” in our model. Thus, the class “Base Element” is used to represent all vocabulary entries, respectively, each term of the vocabulary corresponds to an instance of the class “Base Element”.

The class (Fig. 3) has the following attributes: *ID*, *name*, *subtopic*, *supertopic*, *description*, *synonyms,* different *translations (e.g. german, spanish*, *italian, dutch, french), selectable* and *accepted.*

The attribute *ID* holds the identification number which is automatically assigned by Protégé. The attribute *name* holds the describing name of the vocabulary item itself. The parent-child relation between several terms is modeled using the *sub-* and *supertopic* attributes. The values of both attributes have to be other instances of “Base Element”. To ensure validity of the classification and to make the work of the domain tagging experts easier the model is extended with an inverse relation so that a *subtopic* relationship automatically induces a *supertopic* relationship and vice versa. For example, if “cinema” is added as *subtopic* to “recreational facilities”, “recreational facilities” will be add as *supertopic* to “cinema”.

The attribute *selectable* has a Boolean value and specifies if an item of the vocabulary is selectable for classification purposes or not. For example, it is not meaningful to label a learning resource like a picture of the Rocky Mountains with “landforms”. Instead, one or more specific landforms, like “hills” or “mountains” have to be chosen. Therefore, the *selectable* attribute identifies terms in the vocabulary that are used for structuring the vocabulary but that should not be used for tagging purposes directly.

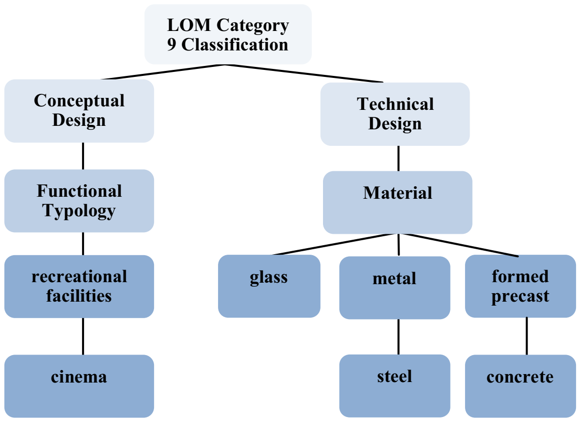

Fig. 2: Excerpt of the MACE vocabulary

The attribute *description* holds a free text which specifies the item. A value of the attribute *translations* contains an additional tag representing the language of the translation, e.g. “it”, “de”, “nl”, “es” or “fr”. The attributes *synonyms* and *translations* ensure that a learning object, which is tagged with a specific item, also can be found if a user searches with a synonym of this term or in a different language. For example, the term “insulate glass”, a *subtopic* of “glass”, has the *synonym* “insulated glazing”. Consequently a user can find a learning object which is tagged with “insulate glass” even when he uses the search term “insulated glazing”. Currently, these attributes have to be filled in manually because the terms are too domain specific to use an ordinary translation tool.

The attributes *name*, *supertopic* and *selectable* are required for each item, all other attributes are optional. In order to foster the discussion needed for the extension of the vocabulary, we introduce the Boolean attribute *accepted* and the two utility classes "Annotation" and "Change". The *accepted* attribute identifies terms that agreement has been achieved on with a "true" value while those elements where discussion is still necessary have "false" as value for their *accepted* attribute. The term and also the children are ignored in all MACE applications until the value of its *accepted* attribute is set to "true".

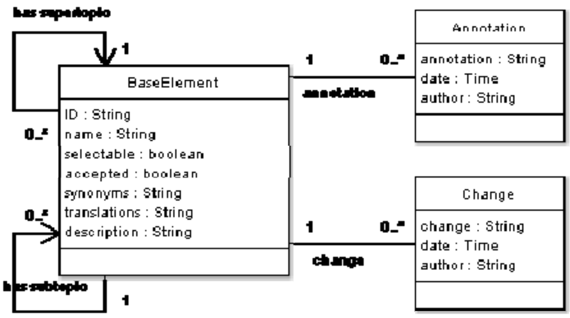

Fig. 3: UML diagram

The class "Annotation" captures comments and annotations, when they were made (attribute *date*) and by whom (attribute *author*). In addition, the "Change" class captures the changes that occurred on the respective term, when they were made and by whom. While our approach does not enable a full-fledged collaboration, as e.g. suggested in [14], it does suffice for the needs of the maintaining body.

As depicted in Fig. 4, the instances of "Base Element" are listed in the Instance Browser window when the class "Base Element" is selected in the Class Browser window. The attribute values of an instance can be edited in the Instance Editor window.

4. Vocabulary maintenance

In the following chapter we will outline how the collaborative work process is supported using Protégé and how we used and restructured the plug-ins to suit our needs.

4.1 Maintaining the vocabulary as group process

Our main objective is to support the collaborative process of vocabulary maintenance in MACE and to ensure the validity of the vocabulary. The MACE vocabulary is maintained by a group of domain experts, architects here.

In order to introduce a change into the vocabulary, they need to come to an agreement on the change. Such changes happen e.g. when new repositories should be integrated. Additionally, expert taggers and registered end users are not only allowed to assign keywords from the MACE classification using the dedicate widgets, but also to propose new free keywords. The most reoccurring new keywords will be evaluated during periodical sessions. The accepted keywords will be integrated in the MACE classification.

To support this process we extended the vocabulary model as described above. Additionally, the discussions can take place within Protégé using the discussion threads and by adding notes to terms.

We discourage the deleting of vocabulary entries in order to ensure correctness within the MACE metadata repository. We strive to avoid a tight coupling of the metadata repository with the vocabulary service. Therefore, if a term is deleted from the vocabulary, we would need to modify the respective learning resource metadata sets in the metadata store. In order to avoid this tight coupling, we do not delete any terms from the vocabulary, but set the value of the attribute *selectable* to "false".

An item which is not selectable and has no children, resp. an item which is not selectable and has only children which are also not selectable is not shown in the tools which are using the vocabulary i.e. the enrichment tool. Consequently, it is possible to delete whole branches of the vocabulary without losing information.

A similar approach is used for merging two terms. First, one term has to be deleted which means *selectable* gets the value "false", the children of this term get the second term as new *supertopic* and the deleted term is added as *synonym.*

Because there are many persons who need access to the vocabulary, but not all these persons are allowed to change it, we define different groups like developers and readers, where the developers are allowed to change the vocabulary and the readers have only the permission to read the vocabulary. Every user can be part of one or more groups and gets the associated rights automatically. This is done using the Protégé user management.

4.2 Collaborative use of the Protégé Plug-Ins

4.2.1 Data maintenance. We transformed the original MACE vocabulary from a JSON file (used for the tagging tool) to an XML file using a specifically created translation tool. The XML file is then imported into Protégé via the XML Tab Widget. Instead of the default pprj storage file, we use the mySQL database as backend to Protégé.

For the internal management of Protégé, each instance of the class “Base Element”, which means each vocabulary entry, gets an automatically generated ID. If a learning object gets tagged with a term of the vocabulary, we use this ID instead of the *name* to identify the specific vocabulary entry, so that it is possible to change the *name* of an instance without losing information.

4.2.2 User access. There exists an outdated web API for Protégé with only a few functionalities [8]. Because of the urgent need for an extended web access API, the community is developing a new one, but in the meantime a compromise must be found. It is possible to store the vocabulary on a server and to install the Protégé tool local on the user’s computers. The users have to start the program locally and to connect to the server to get access to the stored vocabulary. In order for multiple users to work on the classification at the same time, the multi-user-mode has to be selected (see 4.2.3 Collaborative Protégé).

4.2.3 Collaborative Protégé. Collaborative Protégé [4] is an extension of Protégé providing a multi-user mode to support simultaneous access and editing of the ontology while all changes made by one user are seen immediately by all other users. All changes are tracked and are shown in the “Collaborative panel” (see Fig. 4). Additionally there are possibilities to discuss the changes in Protégé via discussion threads or through the chat functionality.

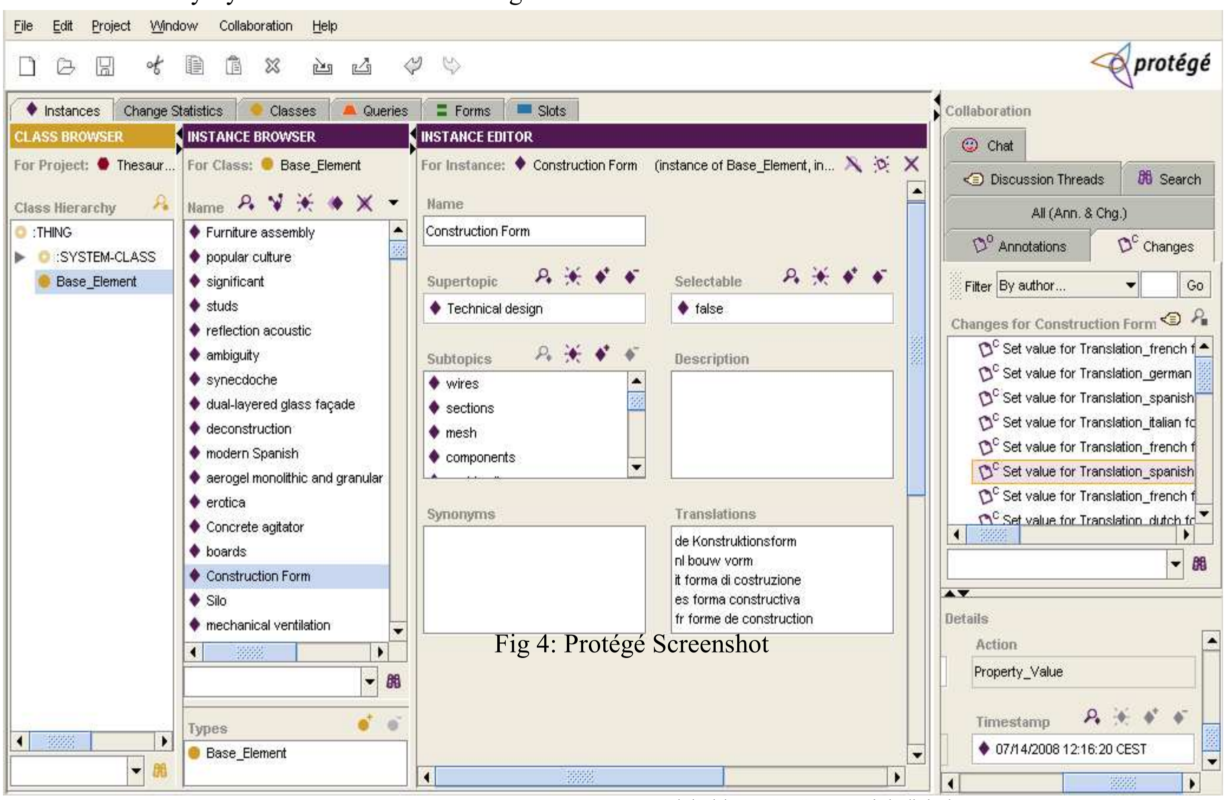

Fig 4: Protégé Screenshot

4.2.4 Web Service API. The web service interface to Protégé, named Neuron, is not supported anymore and works only with an old Protégé server, but provides a lot of functionalities we need. It is i.e. possible to change the classification via the web service API. Unfortunately, changes done through this interface are not tracked. Hence we simply use read-only functions like “get all items”.

Additionally we created a second web service, named Vocabulary Service, which invokes Neuron and returns the MACE vocabulary in the specific XML structure that is need for some MACE applications (see Fig. 5 for an excerpt of the XML output).

5. Experiences and Conclusion

The Protégé tool is used by a group of 20 architectural experts. They are already working in the area of architectural classification. But most of them have no deep computer science background nor experience in the computer supported creation and maintenance of architectural vocabularies.

<item id="id4">
<label lang="en">Material</label>
<selectable>false</selectable>

```
  <Children>
    <item id="material800">
      <label lang="en">glass</label>
      <label lang="de">glas</label>
      <selectable>true</selectable>
      <Children>
        <item id="material1010">
          <label lang="en">insulate glass </label>
          <label lang="en">insulated glazing </label>
          <selectable>true</selectable>
        </item>
      </Children>
    </item>
  </Children>
</item>
```


Fig. 5: Excerpt of the XML output

Our first experiments with this group of architects already show that most of our requirements are satisfied. With the help of a face-to-face introduction to the tool, as well as online tutoring, the architects learned quickly how to use Protégé. The feedback gathered from these meetings through oral examinations and observations indicates that the model of the classification used in Protégé as well as the tool itself proves very suitable.

8964 instances of the vocabulary were already used to tag about 2500 learning instances with meaningful keywords. Also, about 300 terms are already manually translated into other languages thus introduce multilinguality for classification purposes.

Nevertheless, the Protégé tool cannot satisfy all our requirements. It does not provide a simple back-up method in multi-user mode. Instead, the Protégé server has to be stopped and the backup program has to be started locally on the server in standalone mode, afterwards the Protégé server has to be restarted. Therefore, only the administrator can create back-ups thus creating an unnecessary bottle-neck in multi-user ontology management. Furthermore, the architects need to install the Protégé client software locally which creates some management overhead and fault resolution. Therefore, we would prefer to use a web frontend instead of locally installed programs.

6. References

[1] Duval, E. (Ed.): IEEE Standard for Learning Object Metadata 1484.12.1-2002, IEEE 2002

[2] Stefaner, M., Dalla Vecchia, E., Massimiliano Condotta, Wolpers, M., Specht, M., Apelt S., Duval, E.: MACE – enriching architectural learning objects for experience multiplication. In: Duval, E., Klamma, R., Wolpers, M. (eds.) Creating New Learning Experiences on a Global Scale, LNCS, vol. 4753, pp 322--336 Springer, Heidelberg (2007)

[3] http://www.getty.edu/research/conducting_research/

[4] Tudorache, T., Noy, N.: Collaborative Protégé. In: Social and Collaborative Construction of Structured Knowledge Workshop held at WWW 2007 (2007)

[5] Protégé, http://protege.stanford.edu/

[6] http://www.ascinfo.co.uk/9/category/category13_9.html

[7] http://interface.mace-project.eu/tools/enrichment/?contentId = oai:winds.gmd.de:16391.1

[8] Protégé Web Browser, http://protegewiki.stanford.edu/index.php/Protege_Web_Browser

[9] Apollo CH http://apollo.open.ac.uk/

[10] http://www.ksl.stanford.edu/software/ontolingua/

[11] Abu-Hanna A, Cornet R. On the Correspondence between Protégé and Conceptual Frameworks for Understanding Terminological Systems Published in: Proceedings of the 6th Protégé Workshop, Manchester, United Kingdom, 2003.

[12] Christian Prause, Stefaan Ternier, Tim de Jong, Stefan Apelt, Marius Scholten, Martin Wolpers, Markus Eisenhauer, Bram Vandeputte, Marcus Specht and Erik Duval, Unifying Learning Object Repositories in MACE. First International Workshop on Learning Object Discovery & Exchange (LODE 2007) at EC-TEL 2007, Crete, Greece, September 2007

[13] Neuchâtel Terminology Model - Part I: Classification database objects types and their attributes (Version 2.1) (Neuchâtel Group), Part B of Common Metadata Framework, published online at the United Nations Economic Commission for Europe website: http://www.unece.org/stats/cmf/PartB.html

[14] Aldo Gangemi, Valentina Presutti, Carola Catenacci, Jos Lehmann, Malvina Nissim. CODO: an OWL metamodel for collaborative ontology design, Workshop Proceedings of the World Wide Web conference 2007, Banff, Canada

[15] H.Chen "Collaborative Systems: Solving the vocabulary problem". IEEE Computer, Pages 58-66, May 1994