Enterprise Ontology based Artefact Management

Andreas Billig, Kurt Sandkuhl

School of Engineering
Jönköping University
P.O. Box 1026
55111 Jönköping, Sweden
kurt.sandkuhl@jth.hj.se
andreas.billig@jth.hj.se

Abstract: Structuring enterprise information and supporting knowledge management is a growing application field for enterprise ontologies. Based on an industrial case from automotive supplier industries, the paper proposes the use of an ontology for artefact management in engineering of dependable systems, illustrated by the tool ArtefactManager. Navigation and search in the Artefact-Manager is performed using categorizations of artefacts by means of so-called taxonomic paths through the enterprise ontology. One of the main challenges is to permit evolution at least for those parts of the ontology which are used for categorizing artefacts. This paper introduces an approach to support the evolution process on the taxonomy level in conjunction with the level of categorizations.

1 Background

Many engineering disciplines heavily use documents to capture requirements, specifications, design decisions, assembly instructions, test procedures or other artefacts contributing to development and design processes. Although there is a clear trend towards model-based development, i.e. using model-based representations of all artefacts in an integrated tool chain, reality in a lot of enterprises continues to be characterized by document management, document retrieval and the struggle for keeping related documents consistent. A document in this context denotes – according to the international standard ODA [ISO86] - a structured amount of information that is meant for human perception. An artefact is – according to [Lar04] -the general term for all work products in an engineering process, many of them being represented as documents.

Support for managing documents has been subject of research since decades in fields like document engineering, electronic publishing, hypertext & hypermedia or semi-structured documents. Progress in semantic technologies and their use in knowledge management systems or web-portals create new possibilities for easing document management and integration.

Main intention of this paper is to contribute to research in the field of artefact and document management in engineering fields. The aspects considered include (1) the use of taxonomic ontology paths to express categorizations of artefacts, (2) evolution of these categorizations triggered by changes in the ontology and (3) required functionality of a web-based tool for document management.
The next chapter will introduce an industrial case motivating ontology-based document management. Chapter 3 will introduce the tool developed for artefact management. In chapter 4, focus is on evolution of the taxonomic part of the ontology. Chapter 5 summarizes the findings and gives an outlook on further research.

2 Industrial Application Case

The application case motivating research on ontology-based artefact management is a Swedish automotive supplier developing software-intensive systems. Manufacturers and suppliers in automotive industry today are facing an increasing set of customer requirements, an increasing need of integration of different electronic components to build complex functionality, and increasing safety requirements. This leads to a large variety of software components for dependable systems in automotives. Challenges in this field are (1) to store and reuse established components in such a way that they do not hamper the creativeness of the engineer, and (2) to manage the engineering solutions and components efficiently for different views.

The case was part of the SEMCO project aiming at developing methodologies and concepts that contribute to meet these challenges. The project included a detailed study of dependable system development at the automotive supplier, which was based on (a) a description of the supplier’s software development process with defined procedures for development and management of software, and (b) material from two example cases for requirement handling. The first case included customer requirements, functional requirements derived from the customer requirements, and HW/SW requirements derived from the functional requirements. The second case included only hardware/software requirements as these requirements were delivered directly to the supplier. Furthermore, interviews with several members of the software development department were conducted.

Based on this study, the project developed a frame concept for integrated model management, which is based on two main elements

- Feature models for capturing commonalities and dependencies of versions and variants in the automotive supplier’s product families [Ka90, STW07]
- An enterprise ontology for capturing knowledge regarding products, development processes and organisation structure [BOS06]

The two main elements are used in different methods and technology support, including

- a domain repository supporting model integration [SB07]
- a tool for artefact management for supporting management of documents and artefacts in the development process (focus of this paper)

Furthermore, an approach for semi-automatic construction of ontologies was developed and used for constructing for the automotive supplier. The intended use of the enterprise ontology was to capture correspondences between the different artefacts, express overall constraints and provide navigation support between the artefacts for different stakeholders. The resulting ontology consisted of 379 concepts.
3 Artefact Manager

Within the application context described in the previous chapter, development of a prototype system for ontology based artefact management was started. Main intention of this development was to explore the possibilities of using taxonomic paths for categorizing documents and to investigate feasibility of such an approach. The prototype of the "ArtefactManager" (AM) was developed as part of a final thesis work [RW06] and is briefly described in this section. In the context of the AM, an artefact is represented by three elements: meta-data captured in attributes, categorization captured in taxonomic paths (cf. section 4), and reference to the actual document.

The AM provides functionality for creating different artefact types and different attribute types. Categorization of an artefact is performed by selecting paths within the taxonomic structure of the enterprise ontology. Besides functions for creating, modifying and deleting artefacts, artefact types or attribute types, the AM allows for two different types of searching: searching the attributes and searching based on the taxonomic structure. In the latter case, the user selects concepts in the enterprise ontology. This set of concepts is used for searching the categorization of the stored artefacts in order to identify matching documents. Depending on the number of matching concepts, a degree of similarity is calculated.

The AM was implemented as tab-widget plug-in for Protégé. A tab-widget-plug-in is a plug-in which gets its own tab in Protégé, it has access to the knowledge base of
Protégé and can show graphics like panels from the Java Platform or Class-selectors from the Protégé class library. For viewing ontologies a class called MetaDataPanel has been used. It also supports highlighting of concepts or concept paths. The artefact manager tool was presented to industrial users in order to illustrate the potentials of enterprise ontologies in artefact management. Furthermore, the tool was used in an academic setting for testing of ontology matching approaches. Figure 1 shows the graphical user interface of the tool.

Differentiation between logical structure, layout structure and meta-data for describing content, presentation or both has been established in document engineering already in the 80’s (cf. [Fu89]). A prominent example for meta-data is the Dublin Core Standard.\(^1\) This principal structure is valid even for contemporary document management applications and is complemented by navigation structure or time-related information. In the context of this paper, the way meta-data is stored is of specific interest. Two principal ways can be distinguished:

- Embedding meta-data in the document, i.e. meta-data are part of the document model,
- Managing meta-data separated from the document in the management systems.

Most contemporary artefact management solutions use a hybrid approach, as file formats contain a core set of meta-data for specific purposes, which are complemented with additional information on system side. Although the use of ontology-based meta-data is a rather new approach, we can already see the same two types of approaches:

The PDFtab proposed by Eriksson [Erik07] embeds ontologies into the structure of PDF documents and uses these ontologies for navigation support. Within the knowledge portals, like BaSeWeP [BGS05], the ontology fragments describing the content of the documents are stored in the portal, i.e. separated from the documents.

The proposed ArtefactManager tool follows the latter approach. In comparison to earlier work, we do not use semantic nets but the taxonomic structures included in the ontology for categorizing the content.

### 4 Artefact Categorization and Evolution

This section will describe the structure of categorizations and the underlying parts of the enterprise ontology in more detail and discuss their evolution. The parts of the enterprise ontology which are used for categorizations are restricted to its taxonomic structure which is restricted to be acyclic.

Figure 2 shows examples of such taxonomies from the area of system development in conjunction with training courses for employees of the enterprise. E.g., taxonomy a) consists of the concepts training prerequisites (TP) and training goals (TG) at the top level. These concepts subsume the concept competence (COMP) that in turn subsumes system administration and development competences (ADMIN, DEV). Taxonomy b) has in addition social competences (SOC) which subsume team skills (TEAM). The semantics of subsumption is defined as usual, i.e., every competence is a (potential) training goal.

\(^1\) [http://www.dublincore.org/documents/usageguide/](http://www.dublincore.org/documents/usageguide/)
Given such taxonomies, categorizations of artefacts can be defined. The usual way to categorize artefacts would be to define a subset of the concept set which constitutes the taxonomy. More precisely, classical categorization can be viewed as a mapping from the set of artefacts to the power set of the concept set occurring in the taxonomy. From the viewpoint of ontology languages, this corresponds to assigning concepts to instance properties, namely artefact instances, in the sense of OWL 1.1.\(^2\) For the sake of clarity, we do not consider instances together with the type relation here. Note that incorporating instances without using the type relation can easily be achieved by using singleton concepts.

Going beyond classical categorization, we want to make use of the expressiveness stemming from the multiple is-a-relationships within the categorization process. Our approach allows for contextualization by defining categorizations not only with concept sets but also with concept paths through the taxonomy (taxonomic paths). Basing on the directed graph representation \(G_T\) of a taxonomy \(T\), a taxonomic path is defined as an acyclic path through \(G_T\), i.e., a sequence of concepts along the subsumes-arc. From the technical point of view, our approach allows for building context terms over ontology concepts not only during ontology construction. More precisely, assuming that \(TP\) denotes the set of all taxonomic paths, categorization is now defined as a mapping from the set of artefacts to the power set of \(TP\). E.g., let \(A\) be an artefact describing the prerequisites of a training course and let \(B\) be an artefact describing the learning goals of a course, then the path set \(\{(TP, COMP, TECH, ADMIN)\}\) and the path set \(\{(TG, COMP, SOC, TEAM)\}\) from taxonomy \(c\) could be a categorization for \(A\) and \(B\), respectively.

One special issue in our application setting was the fact that at least the taxonomic part of the enterprise ontology was subject to modifications driven by users as well as ontology developers. In order to keep categorizations consistent with respect to the taxonomy, we have to consider the evolution process on the taxonomy level in conjunction with the level of categorizations. In general, our evolution process can be described as a relationship between two taxonomies and sets of categorizations which is shown schematically in the following diagram.

with taxonomies \( T, T' \) and valid categorization sets \( C, C' \). Of course there are several possibilities which elements are given and which can be derived. In our application setting we favour the following evolution process: the taxonomic part \( T \) of the enterprise ontology and the set \( C \) according to it is given. Furthermore we assume that \( T \) has been modified applying operations \( e \). Then evolution support means to derive operations that modify \( C \) accordingly, i.e., to derive \( e' \) and \( C' \). We have developed three basic operators for modifying taxonomies, namely (i) \textit{CreateSubConceptRel}, (ii) \textit{DeleteConcept}, and (iii) \textit{ChangeSubConceptRel}. An operation only succeeds if the taxonomic part stays acyclic. We ensured the existence of operation sequences for every possible pair \((T, T')\).\(^4\) Furthermore these operators cannot produce disconnected graphs which is essential for building categorizations in our context. An example of applying these operators is shown in Figure 2. The transitions between a-b, b-c, and c-d apply operator (i), (ii), and (iii), respectively.

As described above a categorization is a set of concept paths through a taxonomy. For the structure changing operations (ii) and (iii), corresponding path operations on the categorization have to be performed so that the resulting concept paths are valid wrt. the new taxonomy.\(^5\) The generation of a sequence of path operations based on a net operation \( e \) is denoted by \( G(e) \). Finally the whole evolution process according to the diagram above can be characterized by a number of evolution steps where every step derives \( e' = G(e) \) and \( C' = e'(C) \) with given \( T, C, e \) as input.

5 Summary and Future Work

Based on the context of an industrial case from automotive supplier industries, the paper proposed the use of an enterprise ontology for artefact management in engineering of dependable systems with a focus on requirements engineering and feature modelling. The enterprise ontology captures knowledge about the development process (activities, work results), product features (commonalities, dependencies, structure), organisation structure (roles and hierarchy). An artefact manager tool is presented offering the management both, attribute-based meta-data and categorization meta-data and references to the actual artefact. The tool is implemented as Protégé plug-in and stores attributes, categorization as additional parts of the enterprise ontology.

The main contributions of this paper to the research field are to introduce and discuss the use of taxonomic paths for categorization of artefacts in an engineering context. Furthermore, the evolution of these taxonomic paths is investigated and discussed. Main

---

\(^3\) Validity means that the concept paths from \( C, C' \) are paths over \( T, T' \).

\(^4\) We assume that every taxonomy has a root concept.

\(^5\) A detailed algorithmic description which also has to deal with ambiguities is beyond the scope of this paper.
limit of research so far is that there are no experimental results from applying the AM in real world engineering scenarios.

Future work will consider the extension of the evolution process to non-taxonomic parts of the ontology. Furthermore improvement of matching categorizations in order to determine semantic similarity between artefacts will be investigated.

Acknowledgement

Part of this work was financed by the Swedish Knowledge Foundation (KK-Stiftelsen), grant 2003/0241, project SEMCO, and by the Swedish Institute, grant 01215/2007, project CORELIB. The authors would like to thank Alexandre Walter and Ragnvald Rundberg for the implementation of the ArtefactManager tool as part of their final thesis work.

References


[RW06] Ragnvald Rundberg, Alexandre Walter: Ontology-based management of artifacts in model-based software engineering. School of Engineering at Jönköping University, April 2006
