Adaptive Knowledge Management: A Meta-Modeling Approach and its Binding to XML

Christian Süß

Fakultät für Mathematik und Informatik
Universität Passau
D-94030 Passau, Germany
suess@fmi.uni-passau.de
http://daisy.fmi.uni-passau.de/

Abstract

In this paper we propose a meta-modeling approach to adaptive knowledge management. It extends previous work by introducing an application-specific layer which allows to specify meta-models for different types of application such as teachware or business information. These models focus on the conceptual content structure of knowledge modules and their modular composition. They facilitate their managing, exchanging and dynamical composition to personalized information spaces. We further introduce the concept of view specifications which allow the adaptation of knowledge modules to the individual user. As an instance of our generic framework we discuss a teachware-specific meta-model and its binding to the XML based Learning Material Markup Language $LMML$.

1 Introduction

Today, business information, teachware or knowledge in general is frequently provided on an intranet or on the internet in a variety of formats that is more or less unstructured and can essentially only be accessed using the corresponding viewers or readers. Users often vary in their interest and are seeking an individual access to these knowledge modules by adaptable navigation and presentation of given contents and structures [4] in different types and domains of application. Authors usually want to find and combine knowledge modules. They also want to adapt them to new audiences at various levels of granularity. Both not only need general hypermedia modeling concepts [8] but also application-specific and domain-specific structures and meta data. Unfortunately, in most cases, meta data which could be exploited by a knowledge management system is either entirely missing, individually assigned in a rather ad hoc way, or describes learning objects [7] or electronic resources [5] only at the coarse granularity of large chunks of knowledge.

We present a meta-modeling approach to adaptive knowledge management which extends previous work [16] by introducing an application-specific layer. It allows to describe the modular structure and the *conceptual document structure* of knowledge modules in an application-specific way which is needed in composition, navigation and adaptation. The latter also profits from the new concept of view specifications which specify transformations on semi-structured knowledge modules [1]. Our approach does not depend on a relational or object-oriented modeling of an entire topic itself and thus avoids the huge effort needed in conceptually modeling an academic web site [11], electronic product catalogues [6], the domain of multimedia [18] or programming languages [10].

The rest of the paper is organized as follows: In section 2 we introduce our meta-modeling architecture. Section 3 describes our abstract meta-model and its sample instantiation for teachware applications. In section 4 we discuss the binding of our models to XML. The paper is concluded with a summary in section 5.

2 Modeling at Different Levels

The requirements described in the previous section emphasize the need for a common conceptual and modular structure of knowledge, while at the same time they call for application-specific instantiations, e.g. for teachware or business information which themselves can be further specialized by domain-specific models. This suggests to use a meta-modeling approach to knowledge management rather than to focus on single models for each type and each domain of application.

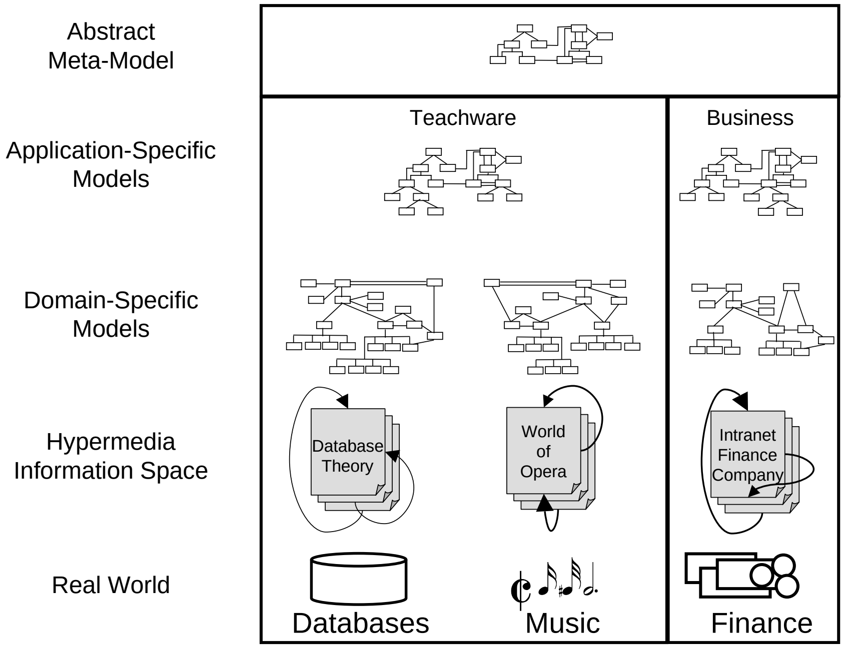

Figure 1: Modeling at different levels

The bottom layer of our meta-modeling architecture (figure 1) consists of the *real world* knowledge to be managed. At the second layer, *hypermedia information spaces* are describing the given domain of application. Their conceptual and modular structure, i.e. the admissible *form* of the hypermedia documents and structures is described in *domain-specific models*[16]. In the *application-specific* layer the common structures of e.g. teachware or business information is specified. Finally, the common *abstract meta-model* describes what it means to be a application-specific model, i.e. gives a definition of the general kind of structure description that is accepted and can be understood by a knowledge management system.

3 Abstract Meta-Model and its Application-specific instantiations

The properties found to be common to knowledge coming from different types of application are realized in a straightforward way in the meta-model (cf. center of figure 2) which can for example be instantiated in a teachware-specific way.

3.1 Modeling the Conceptual Structure

On the right hand side of figure 2, the abstract conceptual content structure of the content provided e.g. by teachware or by the business information in a company's intranet is specified. Our approach does not

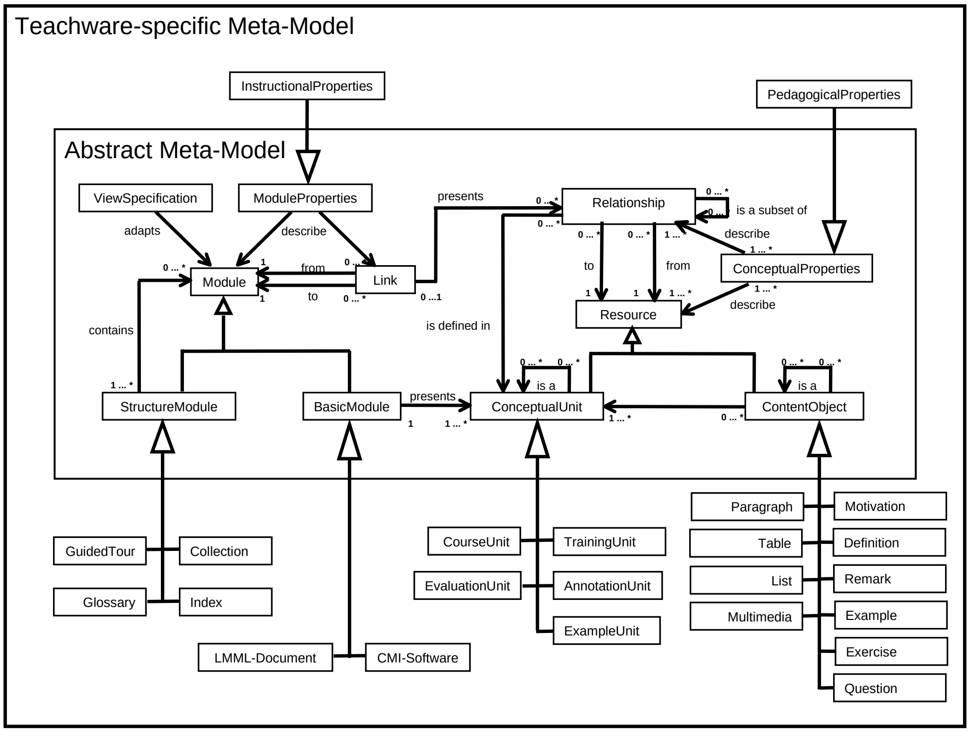

Figure 2: Abstract meta-model and its teachware-specific instantiation

depend on the conceptual modelling of a domain itself. Thus, the smallest units of these contents are not the single subjects or topics but so called *ConceptualUnits*. Their variable inner structure is realized by *ContentObjects*. Both concepts are, at the meta-model level, instances of the general concept *Resource* thus allowing all kinds of relationships, described by *Relationship*, to hold for *ConceptualUnits* and *ContentObjects* as well. To allow the assignment of different types of metadata to *ConceptualUnits*, *ContentObjects* as well as *Relationships*, we use the concept *ConceptualProperties*. Figure 2 also shows the teachware-specific instantiations of these concepts. In teachware there are the *ConceptualUnits CourseUnit*, *TrainingUnit*, *EvaluationUnit*, *AnnotationUnit* and *ExampleUnits* which can be further specialized to domain-specific units like a *SQLTrainingUnit* in the domain of teaching and learning database theory. In business information there are *DepartmentUnits*, *EmployeeUnits*, *ProductUnits* etc. which can be further specialized to *FinanceProductUnits* in the domain of finance business information. These *ConceptualUnits* contain the different *ContentObjects Motivation*, *Definition*, *Remark*, *Question*, *Exercise* and *Example* which can contain the semantically unspecified floating objects *Paragraph*, *Table*, *List* and *Multimedia*. To increase readability we have omitted some details in figure 2. For example, our teachware-specific meta-model also specifies which *ConceptualUnit* contains certain *ContentObjects* or which *ContentObjects* are involved in a particular *Relationship*. In the domain-specific model for teaching mathematics or computer science [16], these *ContentObjects* could be further specialized to *Theorems*, *Proofs* or *Algorithms*. In business information *ProductUnits* might contain *Descriptions*, *TargetAreas* or *Suppliers*. For the time being we do not introduce standardized teachware-specific types of relationships but keep this additional feature for future use. At the moment, *Relationships* like *explains*, *illustrates* or *trains* are rather defined implicitly by their corresponding source types and target types as in *Exercise trains Definition*. Finally, teachware offers *PedagogicalProperties* as instance of *ConceptualProperties*.

3.2 Modeling the Modular Structure

On the left hand side of figure 2, the abstract modular structure of hypermedia information spaces presenting teachware, business information or knowledge in general is specified. *ConceptualUnits* are presented to the user by different kinds of *BasicModules* which are the smallest units and the terminal nodes of a polyhierarchical hypermedia structure. Each *Module* has to be contained in at least one *StructureModule*. The meta-model also specifies which *Relationships* are relevant to navigation and therefore are presented by *Links* which can be instantiated with different kinds of operational behaviour. All modules provide information necessary for composition, navigation and adaptation. To allow the assignment of different types of metadata to *BasicModules*, *StructureModules*, and *Links*, we use the concept *ModuleProperties*. Finally, each *Module* has a *ViewSpecification* either directly assigned or inherited from its supermodule, which specifies transformations used in adaptation (cf. section 4).

Looking at the modular structure of teachware, a *CourseUnit* can be presented by a $LMML$ *document* (see section 4) or a *CMI-Software* component, e.g. a Macromedia Director Movie or an Asymetrix Toolbook book [3]. These *BasicModules* are grouped by *StructureModules* which realize the multiple teaching strategies [17] *GuidedTours*, *Collections*, *Glossaries* and *Indexes*. In teachware, *Links* between modules of the same *GuidedTour* are called inner links whereas *Links* crossing structure boundaries are called cross links. The containment relationship *contains* of submodules in modules can be used to switch between different learning contexts. Finally, *GuidedTours* provide additional links for sequential navigation of their submodules.

4 XML Binding

A knowledge management system should be able to manage knowledge *Modules* from different types and domains of application using database technology which is particularly suited to improve especially the access to huge amounts of documents [2]. To make use of the advantages of our meta-modeling approach presented in the previous sections, we need a format which allows storage, easy access, combination and adaptation of *Modules* by a knowledge management system. As an XML (Extensible Markup Language, [9]) document is a hierarchy comprising elements that have contents and attributes, XML is perfectly suited for representing our hierarchical modeling, i.e. the conceptual content of the basic modules as well as the modular structure. Our teachware-specific model and its domain-specific instantiations serve as a well defined basis for a corresponding XML based markup language, the learning material markup language ($LMML$) [12], the elements of which represent conceptual *ContentObjects* by syntactical means. Furthermore, *ViewSpecifications* are realized by Style Sheets of the Extensible Style Language (XSL) [9]. They use *ModuleProperties* and *ConceptualProperties* coming from XML bindings of different metadata proposals and standards like [7] for restructuring and rendering the corresponding $LMML$modules for different audiences, layouts or platforms.

5 Conclusion

A meta-modeling approach to adaptive hypermedia-based knowledge management has been presented. It extends previous work by introducing an application-specific layer which allows to describe knowledge about aspects of the conceptual and the modular structure of teachware, business information etc. We have described teachware-specific instantiations which can be further specialized e.g. by a domain-specific model for course material on database theory [16]. Our approach enables knowledge management systemes like the Passau Knowledge Management System PaKMaS [15] to provide type-specific filtering of *ContentObjects* and *Relationhips*, adaptable presentation and convenient navigation [13] as well as sophisticated query capabilities, the support of automatical integration of new material and the import and sharing of existing material and its composition and configuration to new audiences [14].

Finally we presented a XML binding of our model specifications and discussed how they serve as sophisticated a-posteriori data schemata that allow to apply database technology to knowledge management to improve especially the access to knowledge modules.

References

[1] S. Abiteboul. On Views and XML. SIGMOD Record, Vol. 28, No. 4, 1999/12.

[2] S. Abiteboul, S. Cluet, V. Christophide, M. Tova, G. Moerkotte, and J. Siméon. Querying Documents in Object Databases. *International Journal on Digital Libraries*, 1(1):5–19, 9 1997.

[3] D. Boles and M. Schlattmann. Multimedia-Autorensysteme. LOG IN, Vol. 1, 1998.

[4] P. Brusilovsky. Efficient Techniques for Adaptive Hypermedia. In C. Nicolas and J. Mayfield, editors, *Intelligent hypertext: advanced techniques for the World Wide Web*, volume 1326 of *LNCS*, pages 12–30. Springer, Berlin Heidelberg, 1997.

[5] Dublin Core Metadata Element Set, Version 1.1. http://purl.oclc.org/dc/documents/rec-dces-19990702.htm.

[6] N. Koch, B. Gaede and J. Schneeberger. Rapid Development of Electronic Product Catalogues. Technical Report 9803, Ludwig-Maximilians-Universität München, November 1998.

[7] LOM. IEEE P1484.12, Draft Standard for Learning Object Metadata http://ltsc.ieee.org/doc/wg12/.

[8] F. Garzotto, P. Paolini, D. Bolchini and S. Valenti. "Modeling-by-Patterns" of Web Applications. *Advances in Conceptual Modeling. ER'99 Workshop on the World-Wide Web and Conceptual Modeling*, volume 1727 of *LNCS*, pages 293–306, Berlin, 1999. Springer-Verlag.

[9] C. Goldfarb and P. Prescod. *The XML Handbook*. Prentice Hall, 1998.

[10] N. Henze and W. Nejdl. Adaptivity in the KBS Hyperbook System. In *2nd Workshop on User Modeling and Adaptive Systems on the WWW, May 11th*, Toronto, Canada, 1999

[11] D. Schwabe and G. Rossi. Developing Hypermedia Applications using OOHDM. In *Electronic Proceedings of the 1st Workshop on Hypermedia Development, in conjunction with 9th ACM Conf Hypertext and Hypermedia, Pittsburgh, USA, 1998*, Hypermedia Development Workshop. ACM, 1998.

[12] Ch. Süß. Learning Material Markup Language. http://daisy.fmi.uni-passau.de/pakmas/lmml/.

[13] Ch. Süß and B. Freitag. Meta-Daten als Grundlage für Adaptation in Lernsystemen. In *7. GI-Workshop Adaptivität und Benutzermodellierung in interaktiven Softwaresystemen (ABIS99) Proceedings*, Otto-von-Guericke-Universität, Magdeburg, Germany, September 1999.

[14] Ch. Süß and B. Freitag. Wiederverwendung von Inhalten und Strukturen offener Lernsysteme. In *9. Arbeitstreffen GI-FG 1.1.5/7.0.1 "Intelligente Lehr-/Lernsysteme" Proceedings*, Otto-von-Guericke-Universität, Magdeburg, Germany, September 1999.

[15] Ch. Süß and B. Freitag. Entwicklung und Nutzung von Teachware: Das Passauer Knowledge Management System (PaKMaS). In *Computerunterstütztes Lernen*. Oldenbourg-Verlag, München, 2000.

[16] Ch. Süß, B. Freitag, and P. Brössler. Metamodeling for web-based teachware managment. *Advances in Conceptual Modeling. ER'99 Workshop on the World-Wide Web and Conceptual Modeling*, volume 1727 of *LNCS*, pages 360–373, Berlin, 1999. Springer-Verlag.

[17] Ch. Süß, R. Kammerl, and B. Freitag. A teachware management framework for multiple teaching strategies. In *Proceedings of ED-MEDIA 2000, World Conference on Educational Multimedia, Hypermedia Telecommunications*, Montreal, Quebec, Canada, June 2000.

[18] A. Steinacker, C. Seeberg, S. Fischer and Ralf Steinmetz. Multibook: Metadata for the Web. In *2nd International Conference on New Learning Technologies, Bern, Swizzerland*, pages 16–24, 1999.