Developing Modular and Adaptable Courseware Using TeachML

Dipl.-Inform. Frank Wehner, Dipl.-Inform. Alexander Lorz
Dresden University of Technology
Heinz-Nixdorf-Endowed Chair for Multimedia Technology
01062 Dresden, Germany
{frank.wehner, alexander.lorz}@inf.tu-dresden.de

Abstract: In this paper we present the use of an XML grammar for two complementing projects - CHAMELEON and EIT. Areas of application are modular courseware documents and the collaborative authoring process of didactical units. A number of requirements for a suitable document format will be identified. After comparing existing solutions we introduce TeachML, our XML based document format for adaptable courseware documents and reusable didactical units. We describe how courseware documents can be adapted to the learner' s previous knowledge, aims and infrastructure. Then the process of creating different output documents from a TeachML source will be explained. We continue with considerations about the development process of learning material within a professional publishing environment, leading to the necessity of a virtual repository for storing courseware documents and media assets.

Introduction

In an information society ongoing learning and knowledge updating is crucial, especially in the rapidly growing field of new technologies. Access to online-information from everywhere at any time becomes more and more important, for both private and professional users.

However, it is quite difficult for specialists or educationists to create interactive, multimedia courseware documents because in most cases they lack the necessary skills in the fields of design or programming. Regarding this situation we initiated two projects: EIT (Enabling Informal Teamwork) [1] deals with the informal teamwork processes during the development of courseware documents. CHAMELEON (Cooperative Hypermedia Adaptive MultimEdia LEarning Objects) [2], [3] investigates document formats for modular and adaptable courseware documents. These projects aim at the development of media independent courseware documents for autonomous and tutor-supported learning.

For the description of documents consisting of reusable and adaptable didactical units and pre-structured document templates for frequent teaching situations we started to develop the XML grammar "TeachML" in 1998. The use of XML allows the dynamic generation of output documents in different formats and styles (e.g. XHTML - Extensible Hypertext Markup Language - or TeX) adapted to the user' s knowledge, preferences, aims and computing infrastructure.

Both projects imply a scenario focused on the development of didactical units by professionals who fulfil a certain role in this process (e.g. author, editor, expert of a certain subject, software developer, designer, ..). Therefore the main objective of EIT is to understand how the co-operation of these people can be supported by a web-enabled groupware system and which features are essential for a suitable electronic working environment. For that purpose we will design and implement the prototype of a virtual repository for didactical units and multimedia documents using insights and experiences acquired by converting different kinds of conventional learning material - like on- and offline-media, books and comple-men-tary material - to TeachML. The editor, tutor or teacher can select didactical units from the repository which fit his specific target group or learning situation best to create adapted output documents.

In the next chapter we will provide an introduction to the TeachML document format. Chapter three describes how we are approaching the challenges of the professional authoring process for didactical units in a team-oriented environment.

The TeachML Document Format for Courseware Documents

Analysis of existing courseware material for web-based learning systems reveals that there are primarily two common approaches. The first way is the use of HTML documents, to benefit from the advantages of an open standard (or more precisely a W3C recommendation) to achieve platform independency. Moreover there already exist many authoring tools and export filters for creating HTML documents and it is possible to integrate multimedia elements and interactive behavior using scripting languages, plug-ins, Java applets or SMIL (Synchronized Multimedia Integration Language). However, there are also a number of limitations: Structuring HTML documents whilst preserving adaptability and extensibility is quite difficult. Furthermore HTML is a web-format and not very well suited for creating different output documents – e.g. printed versions or multimedia presentations. Dynamic HTML is very useful for similar looking pages like product presentations, but not for courseware documents.

The second way to create courseware documents is the use of special authoring tools (e.g. Idea [4], Authorware [5]) or universal multimedia authoring tools (e.g. Director [6], Flash [7] or Toolbook [8]). With these tools it becomes easier to develop interactive and media-enriched courseware documents, but the main problem persists: Structuring the documents into reusable didactical units is not supported. Another disadvantage of these tools is the use of proprietary formats which leads to non-interoperable results, making reuse, adaptation and combination of existing courseware modules difficult to impossible.

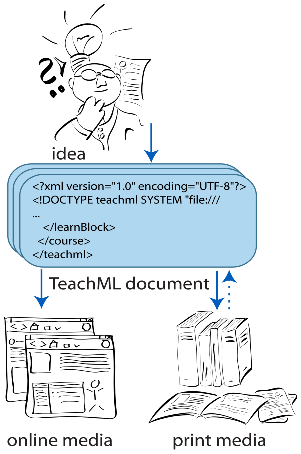
Requirements for a Courseware Document Format

Considering the problems mentioned above some requirements the document format is to meet in order to structure courseware documents will be specified.

- System- and platform-independency must be guaranteed or the necessary tools and interfaces (APIs) have to be available for most platforms.
- The document format has to be extensible in order to integrate custom structures and future technologies.
- Mechanisms for associating the course-structures with all necessary metadata elements have to be provided.
- The format must not enforce the use of specific output formats or presentation platforms.

To meet these requirements we decided to use XML for the development of a document format for courseware documents. XML covers all of the points above: It is independent from a specific platform and operating system and provides the DOM (Document Object Model) as a platform- and programming language-independent API for the development of the necessary tools. XML grammars can be constructed in a way that they easily can be extended by new or alternate elements. There already exists a metadata scheme for teaching resources – IMS (former Instructional Management Systems, now IMS Global Learning Consortium, Inc.) [9] derived from the IEEE LOM (Learning Object Metadata) [10] metadata scheme. Though the IMS scheme itself is independent from a specific document format there is an XML binding that can be used easily.

In the next section we compare two existing markup languages for courseware documents. Afterwards the TeachML document format itself will be introduced. Finally we describe, how TeachML documents can be used to create modular and adaptable courseware documents.



```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE teachml SYSTEM "file:///
...
   </learnBlock>
  </course>
</teachml>
```

Figure 1: Achieving Media-Independency

Existing Markup Languages for Courseware Documents

In the context of the PaKMaS project (Passauer Knowledge Management System) [12] which deals with web-based adaptive hypermedia information systems the document format LMML (Learning Material Markup Language) [11] has been developed. LMML itself is divided into two parts: LMML-structure and LMML-content. The former can be used for structuring courseware content into several units which can be subdivided in basic modules and sections. The latter provides means for representing the real content components. Metadata elements can be integrated into LMML documents using predefined attributes. Thus, there is not much overhead when using metadata. However, integrating a new metadata element requires changes to the LMML DTD. LMML does not allow the definition of different routes through the course and provides no means for modeling dependencies between learner and courseware content.

The TARGETEAM project (TArgeted Reuse and GEneration of TEAching Materials) [13] focuses on supporting the different usage-processes in the lifecycle of courseware material. An XML-based format for courseware documents has been developed recently, called TeachML [14] too. The courseware documents are stored in pools containing modules assembling reusable units. Modules can be structured by nested issue-elements containing text content, tables, links, images and image animations. TARGETEAM-TeachML does not enable the author to create different routes trough or views of the content. Several content elements are defined but there is no way to structure the material in a didactical way. An open question is, why the sub-languages for content elements re-define existing structures for tables (HTML), links (HTML, Xlink) or image animations (SMIL). Java based processors for creating HTML- or TeX-output are used whilst XSLT and CSS should suffice and would be a more open and adaptable approach.

The TeachML Document Format

TeachML has been developed in the context of the CHAMELEON project. Our goals were to give authors the possibility to create reusable didactical units and to enable tutors to select, customize and combine such didactical units to target group specific courseware material.

Each TeachML courseware document is divided into four levels of elements. On the lower level there are *media objects* like text, images, audio, video, formula, reference, but also interactive Java applets, animations, or 3D-scenes. The mediaObject-element hides media-specific properties, e.g. whether an animation is realized with Shockwave or SMIL.

On the second level there are *content units*. Content units group media objects which belong together to transmit their message to the learner, e.g. a figure and its title or a proof which can consist of several formulas, texts and references.

The third level are *didactical units*. They consist of several content units. Didactical units are the primary units of reuse, e.g. example, figure with explanation or different types of exercises.

On the upper level there are so called *didactical structures*. They are used to create real courseware structures. Possible types of didactical structures are course, chapter, chain of arguments, examinations containing several exercises, a overview path or a detailed path. Didactical structure is the only recursive element because high-level structures like a course can consist of lower-level structures like chapters.

Since all relations between media object, content units, didactical units and didactical structures are references all elements must have a unique ID. This ID is also used for another purpose: Every didactical unit can contain an optional list of references to other didactical units which represent didactical preconditions for its use. With TeachML we do not try to build a semantic network - in our opinion this is too costly and in most cases unnecessary effort. However, in difficult and very detailed courses it can help the learner to find the right entry point.

Information about learners can be represented in a compact learner' s model. It contains administrative information but is also tightly coupled with the courseware itself. In order to adapt the presented TeachML course to a certain learner it is possible to specify information about previous knowledge, aims and actual learning success. The previous knowledge and aims of the learner are represented by references to didactical units. For example an aim could be a reference to a set of exercises the learner wants to (or has to) solve at the end of the course in order to get a certificate. The actual learning progress is represented in two ways. The first is a history list with references to the visited didactical units. If some of these didactical units are exercises, information about their solutions will be stored as well. The second way is by using the variables *level of detail* and *level of difficulty*. With this information the appropriate didactical structure (e.g. a path with a matching difficulty) for the learner can be selected.

For assigning metadata to TeachML elements an XML binding for the IMS metadata scheme based on the IEEE LOM scheme is used. IMS defines a core of frequently used metadata elements and a so called standard extension library for other elements. The scheme defines nine categories (e.g. general or educational) and a number of elements inside each category. Every element is optional and can be used in any number and order. However, for an author the great variety of available metadata elements can be quite confusing. Therefore a so called metadata mapping is being defined in TeachML, mapping each TeachML element to the relevant IMS metadata elements along with information about whether the metadata element is mandatory or not. If different types of TeachML courseware documents require different metadata elements it is very convenient to have to update the mapping information only. Some extensions to the IMS scheme became necessary, e.g. for the types of example and argumentation chain. As an equivalent to the existing metadata element difficulty in the educational category the new element detail has been added to this category. All new elements were easily integrated using the extension mechanism of the IMS scheme.

Together with the Cornelsen Verlag [15] we are analyzing how existing courseware material e.g. from schoolbooks can be converted to a TeachML equivalent. This experiment will help us to evaluate the TeachML format in order to find missing types of didactical units and to update or extend existing ones.

CHAMELEON System Architecture

In this chapter we show how TeachML documents can be integrated into the context of the CHAMELEON project. Additionally we describe our current work in this project.

To take advantage of the power of the TeachML document format there is a need for some additional tools. In the last chapter we described, how basic information for adapting a course to the learner’ s previous knowledge and aims can be represented in a learner’ s model. This model can be updated by using information about the interactions between learner and didactical units. Most information is provided by exercises, because they indicate if the learner understood the contents of the course. Exercises should denote a level of difficulty, so this information can be used to make an assumption about the appropriate difficulty for the next courseware documents to be presented.

Exercises are usually interactive components. Thus it is difficult to convert the TeachML based exercise description into an interactive presentation on the learner’ s side. One possibility is to use HTML forms, but this is applicable to multiple choice and completion tests only. So we decided to develop a set of Java applets for exercises. The TeachML document fragment containing the exercise information (e.g. question, true and false answers, feedback information and help topics) is used to initialize the applet. Then the applet dynamically constructs the visual presentation of the exercise, manages the interactions with the learner and returns the learner’ s solutions (and the way he found them).

So far we introduced the TeachML format in order to represent courseware documents, but how can these be presented at the learner’ s computer? Our solution is to use stylesheets to transform the TeachML document into the appropriate output format. This is a quite complex task to perform, because of the large number of different types of content, didactical units and additional information (e.g. from the user model or about the system configuration) which must be taken into account. Currently we are using a combination of XSLT and CSS stylesheets for transforming TeachML documents into an XHTML representation for a web course. In order to provide a printed version of the TeachML document XSL Formatting Objects are used to create PDF documents. In addition an attempt has been made to develop guidelines and stylesheets for adapting TeachML documents to the needs of visually impaired learners based on the W3C recommendation about Web Accessibility Guidelines [16].

The first TeachML based courseware documents were developed using a plain XML editor. Because of the large number of additional information (e.g. IDs, paths and metadata elements) this proved to be a very time-consuming task. Therefore the next step will be the development of a modular authoring tool for TeachML documents and didactical units targeting two main objectives. At first it will simplify the process of structuring the TeachML document as a whole by visualizing dependencies between didactical units and didactical structures. Secondly it will enable the integration of plugin components for creating and adapting didactical units. The main advantage of this approach is that there will be no need for changing the tool framework when the TeachML document format is being extended by new didactical units or existing ones are being changed. That way we will be able to extend the capabilities of the editor by simply adding a reference to the new or changed editor component.

In the next section the authoring process of didactical units themselves will be described in greater detail.

Enabling Informal Teamwork

EIT is a research project in cooperation with Cornelsen-Online and is supported by the German Federal Ministry of Education and Research (bmb+f) as part of the project WEP in the Global Info program [17]. EIT aims at the development of a web-based environment for producing and maintaining courseware documents and learning materials (CD/LM) for various media. The project divides into three parts:

- Unitizing learning material to improve reusability.
- Conceptual design of a virtual repository for web-based multimedia documents and of document management tools.
- Extending these concepts by groupware capabilities with the emphasis on organizing informal collaboration.

Professional Publishing

Publishing companies employ various external and internal specialist for the production of CD/LM. An editor acts as a central role in this process by structuring and coordinating subtasks while authors investigate and create texts, sketches or scripts for multimedia applications. These are transformed into different media by designers, graphic artists, software developers and many others. Coordination of the individual tasks requires additional manual work from the editor who manages the exchange of raw-data and semi-finished materials between the different experts. In this process some disadvantages can be observed:

- The interrelation between tasks, media-assets and meta-information is lost.
- Assets are used in heterogeneous and often incompatible environments. Therefore conversion, systematization and archiving causes additional expenses.
- Reuse and maintenance are impeded by lack oft meta-information about the individual assets.
- Coordination of sub-tasks causes additional work.
- Project status and individual workload are hard to estimate. Bottlenecks are discovered very late, flexible reorganisation of subtasks is difficult.

The vision of an electronic working environment supports modular creation of adaptable and reusable CD/LM for books, on-/offline-media and semi-finished materials. A distributed virtual repository stores media-assets and components for CD/LM together with the associated metadata. Thus an efficient management of the stored information in each stage of its lifecycle becomes possible. Access is provided by a distributed middleware ensuring availability and consistency of the stored data, even if a continuous network connection can not be guaranteed. Different role-specific views of the repository will be provided allowing an easy migration from conventional methods of working and supporting group-aware collaboration.

Research Issues

Currently we are examining professional CD/LM provided by the project partner to improve and validate TeachML. By rewriting these documents we try to find out which processes occur and which metadata have to be filed during the production and maintenance of CD/LM. Simultaneously a modular editor for TeachML documents is designed and implemented. This editor features a plug-in architecture, allowing to extend its functionality to handle special XML sub-structures (e.g. formulas, tables, etc.).

The next step will be the transfer of those development tools and procedures into the authoring process of the Cornelsen Verlag to gather further information about workflows and boundary conditions within a professional publishing process.

The knowledge obtained will be used for the design of a virtual distributed repository for storing TeachML-data and media assets along with the necessary metadata.

Summary

In an information based society knowledge needs to be extended or updated in ever shorter time-spans. Therefore the efficiency of storing information and making it available to the human mind becomes crucial to keep up with this development. Creating modular, adaptable, extendable and reusable courseware materials can be one milestone on the way to meet this challenge. We presented TeachML, an XML based document format for such a class of courseware documents. TeachML enables the courseware author to structure learning material both in respect of content and didactics. Therefore TeachML offers two main concepts: didactical units as units of reuse and combination and didactical structures for creating different views of and routes through the course. A number of didactical units, called atoms, could be identified. Didactical structuress and a compact user model enables us to adapt courses to the learner' s previous knowledge, aims and infrastructure. Because TeachML documents are free of presentation aspects it is possible to create different output documents e.g. an XHTML document for a web presentation and a printable version. The EIT project extends this approach by validating TeachML in a professional publishing environment and by designing a virtual repository for multimedia documents, media assets and associated metadata.

References

[1] EIT project homepage (2001). http://www-mmt.inf.tu-dresden.de/projekte/EIT/

[2] CHAMELEON project homepage (2001). http://www-mmt.inf.tu-dresden.de/projekte/chameleon/

[3] Franze, K. & Wehner, F. (1999). A Document Model for Generating Adaptable Courseware Using Background Data. Proceedings of "WebNet99", Honolulu, Hawaii.

[4] Link&Link Idea product homepage (2001). http://www.linkundlink.de/start_idea.html

[5] Macromedia Authorware product homepage (2001). http://www.macromedia.com/software/authorware/

[6] Macromedia Director product homepage (2001). http://www.macromedia.com/software/director/

[7] Macromedia Flash product homepage (2001). http://www.macromedia.com/software/flash/

[8] Click2Learn Toolbook II Instructor product homepage (2001). http://home.click2learn.com/products/instructor.html

[9] IMS Metadata Specification (2001). http://www.imsproject.org/metadata/index.html

[10] IEEE LTSC Learning Objects Metadata working group homepage (2001). http://grouper.ieee.org/P1484/wg12/

[11] Süß, C. (2000). Adaptive Knowledge Management: A Meta-Modeling Approach and its Binding to XML. In: H.-J. Klein (Ed.), 12. GI-Workshop Grundlagen von Datenbanken, Plön, TR 2005, Christian-Albrechts-Universität Kiel, Germany.

[12] Süß, C. & Freitag, B. (2000). Entwicklung und Nutzung von Teachware: Das Passauer Knowledge Management System (PaKMaS). In: Rudolf Kammerl, Computerunterstütztes Lernen, Oldenbourg, Munich

[13] TARGETEAM project homepage (2001). http://www11.in.tum.de/forschung/projekte/targeteam/index.html.de

[14] TARGETEAM project – The TeachML Language (2001). http://www11.in.tum.de/forschung/projekte/targeteam/extension/html/targeteam4.html

[15] Cornelsen homepage (2001). http://www.cornelsen.de/

[16] Chisholm, W. & Vanderheiden, G. & Jacobs, I. (eds.) (1999). Web Content Accessibility Guidelines 1.0, W3C Recommendation 5-May-1999. http://www.w3.org/TR/WAI-WEBCONTENT/

[17] Global Info project homepage (2001). http://www.global-info.org/