XML-Based Authoring: From Concepts, via Compromises to Applications

Wolf Bubenik, Ivo Hanke, Nadia Juhnke

Freie Universität Berlin
CeDiS – Competence Centre e-Learning / Multimedia
Ihnestr. 24, D-14195 Berlin
{wbubenik | ihanke | njuhnke}@cedis.fu-berlin.de

Abstract

Within the last couple of years, the competence centre for e-learning and multimedia at the Freie Universität Berlin (CeDiS) established a manufacture like production process for e-learning content, which is primarily targeted to large projects, i.e. projects with several authors and an arbitrary volume of content to produce. The most important cornerstones of the production process are an XML document format and an authoring tool for this document format. Unfortunately both were designed only to meet the requirements of two nation-wide projects, which were lead-managed by CeDiS.
The work described in this paper is dedicated to the generalization of that manufacture like production process, especially the development of an adaptable XML document format for e-learning contents and the corresponding editor.
The document format SCDL (Sharable Content Description Language) we specified as XML Schema, is a general document format for modular e-learning content. Besides common features like multimedia integration, it provides a mechanism for deriving project specific document formats from the general format by restriction and not by extension. This mechanism shall prevent that software solutions have to be adapted for any derived document format. Furthermore it fosters the possibilities of re-using and exchanging content.
Based on Microsoft InfoPath we are developing an authoring tool for the SCDL document format. The currently available prototype already provides a comfortable user interface for the authors, which shows a structural, 'semi-WYSIWYG' view of the document. The features implemented so far are sufficient for simple applications, but important components like mathematical formulas and special media elements are still to add.

1 Introduction

CeDiS, the competence centre for e-learning and multimedia at the *Freie Universität Berlin* (FU Berlin), has an experience of almost ten years in producing multimedia e-learning content. These activities started with small and very specialized projects, which aim for a multimedia implementation of an isolated topic. During the last years the focus switched to mass production. Especially the projects *Neue Statistik* ("Neue Statistik", 2005) and *New Economy* ("New Economy", 2005), two nation wide projects for the production of e-learning contents, which are both lead-managed by *CeDiS*, demand for a more standardized production process. This production process was designed to ensure a high degree of consistency of the contents produced by authors distributed all over Germany (Juhnke, Herrmann, 2004). Here consistency has two main aspects: A consistent structure of the content is ensured by separating the curriculum into a set of 'learning modules' and by defining a 'didactic structure' of each of these modules. A consistent layout and design of the contents is guaranteed by mapping the didactic structure to an XML representation and by developing transformations from XML to the desired uniform output representation (HTML, PDF). A key factor for the success of this process was the *Learning Module Editor* (LME), a tool *CeDiS* developed based on *Microsoft Word*. This tool supports the authors to create learning modules which satisfy the didactic structure of the project and which are coded in an appropriate XML representation. The authors can work within their familiar *Word* environment and are not bothered with XML.
Even though this production process was successful for the two projects *Neue Statistik* and *New Economy* it has two main shortcomings, which prevent to offer the process and the tools to other projects: The XML data format of the learning module is not flexible enough to cover projects and contents from very different faculties and authors. Second, the LME is programmed completely using *Word* macros and *Visual Basic* (VBA) and therefore is not stable and especially not fast enough. The conversion from the *Microsoft Word* document model to the XML representations takes about half an hour for a typical learning module.
To overcome these problems we started to develop a more general document format for e-learning content and an editor, which allows authors to create contents satisfying this format in an intuitive and comfortable environment. It should be stressed that this kind of production process is mainly targeted to large e-learning

projects, like *Neue Statistik* and *New Economy*. Such projects are characterized by an arbitrary number of authors at different universities that collaborate in creating a consistent set of modular e-learning contents which shall be recombined to different courses. It is one of the open issues of the work described here, how smaller projects and single authors can also benefit from such a standardized production process. If an XML representation of the contents is not desired for strategic reasons, there are already some authoring tools on the market, which may satisfy the needs of those authors. Another alternative, which also fosters structuring of content and separation of content and layout, is to use a content management system (CMS).
This paper has two main parts: in chapter 2 we present our general document format SCDL and discuss how project specific document formats can be derived. Chapter 3 is dedicated to the authoring tool we currently develop based on *Microsoft InfoPath* for this document format. We close with an outlook.

2 Document Format

A central issue for the development of a document format specification was that the document format has to serve as a common base for authoring, content management and single source publishing.

2.1 Requirements

First of all we have to meet the requirements introduced by the nature of e-learning projects:

- Coherent customization: The heterogeneity of e-learning contents with their various '*didactic structures*' demands for a very flexible and customizable document format. To ease the development of software solutions (authoring tools, content management, transformations) it would be very useful to have a customization mechanism for the document structure that maintains the coherence of all adaptations and extensions. This would also support the sharing of contents among different projects.
- Standard conformance: To ease content export to Learning Management Systems we have to take into account existing standards like IMS-LOM for metadata or QTI for tests.

Further requirements originating from the authoring process include:

- Structure validation: This issue is essential in distributed authoring scenarios with authors using different authoring tools. Throughout the authoring process the authors should be guided to follow the structure defined in the "*didactic model*" of their project. This implies the presence of an abstract document structure definition like a DTD or an XML schema for each project.
- Simplicity: The document structure must be easy to learn and apply for the authors. This implies a flat hierarchy with a minimum number of different content units.

The content management (searching, recombination and exchanging of content units) introduces the requirements for:

- Recombining content units directly leads to a modular document structure with self contained units, implying that all reusable content units contain no cross references to other units.
- The retrieval of content unites is a prerequisite of reusing them. Therefore metadata are needed for all content units that should be reused.

From a single source publishing process we get the additional requirements for:

- separation of content and layout, where we have to consider the demand of the authors to have influence on layout and design of their content
- Alternative resources for one media,
- One general document format to allow the proper rendering of reused content units

Note that the requirements are contradicting in the following issue: We need one general format for all documents and specialized formats for each authoring project.

2.2 Existent XML Document Formats

There are some XML document formats for e-learning contents already available. Most of them are only used within a small community around the developers, so that no candidate for a widely accepted standard seems to be in sight. The document formats we analyzed in detail, <ML>3 ("<ML>3", 2005), XLML ("XLML", 2005) and LMML ("LMML", 2005), all have in common that they are extendable, but do not provide an extension mechanism that maintains the coherence of the extensions.

The SCORM content aggregation model (ADL Technical Team, 2004) is not a document format in the sense discussed here. It defines the aggregation and packaging of content units (learning objects) but gives no description of their internal structure and content.
A document format, which is close to our approach, is DITA (Priestley & Hackos, 2005). It addresses most of the requirements of our authoring process and provides solutions similar to our results. What is still missing is a possibility to incorporate XML contents of other namespaces (mathML, QTI, LOM) ("DITA TC 1.1 issues list", 2005). Unfortunately we became aware of DITA very late and developed our document format completely independent. DITA was not specifically developed for e-learning contents and it has to be checked how it can be adapted for that specific needs.
DOC.BOOK (Walsh, 2002) and the TEI Guidelines (Sperberg-McQueen & Burnard, 2002) are general document formats as well. Their approach is not to provide an extendable and adaptable framework but to define a huge and complex set of mark-up elements in advance. For each purpose an appropriate subset of these elements has to be selected. Because these document formats are mainly designed to meet the requirements of printable output (e-publishing), the support of multimedia contents is limited.
Document formats of conventional word processors (Microsoft Word, Open Office) are not designed to be customized.

2.3 Concepts for a new Document Format

To maintain platform independence of the contents we decided to use XML-technologies. XML is an accepted world-wide standard and various tools and programming libraries for editing and processing XML documents are available. We use W3C Schema for the definition and validation of our document formats and XSLT and XSL-FO to publish in HTML or PDF.
To overcome the contradicting requirement for one general and many specialized document formats we follow a similar approach like the one outlined in the description of DITA (Priestley & Hackos, 2005). We first developed an abstract document structure specification to serve as a common base for all authoring projects. In a second step we define project specific document structures. Here each authoring project is allowed to define its own set of content units. It is required that all project specific document formats are restrictions of this base format. This implies that new types of content units must be restrictions of types described in the base format and the introduction of new attributes is not allowed. This allows document adaptation and extension, maintains extension consistency and facilitates exchangeability.
To achieve conformance to existing standards the incorporation of such contents will be supported. We provide a general element that allows mixed content of any namespace.
Only one notation for all kinds of links and element relations should be used. We choose to apply the W3C recommendation Xlink (DeRose, Maler, Orchard, 2001). Xlink supports simple inline links as well as out of line link collections. The second will be used for all cross references between content units. This approach allows handling cross-references separated from the linked content units keeping those technically context free and exchangeable.
We define a clear hierarchical structure for the contents with three main levels:

- At the conceptual level we provide content units to describe the *didactic structure* of an e-learning project. Like the learning objects of SCORM they are self contained and exchangeable units like 'chapter', 'module' or 'example'. Content units at this level can be specific for each project.
- At the structural level we provide content units for the logical structuring of contents with tables, lists and paragraphs. These units are not necessarily reusable. Here we will have project specific paragraph types as well as common elements like table, list or a common paragraph or title.
- At the media level we provide all the media elements that contain the content. Each media element may contain various resource elements and link target elements. Most of the media elements will be common to all projects.

Metadata will be supported on both ends of the content hierarchy (conceptual units and media elements) to allow reuse of content at these levels.

2.4 Compromises

Some XML-parsers (for example MSXML5.x) do not evaluate W3C-Schema correctly that have been derived from a base schema using schema restriction. Therefore we choose to define the project specific document formats without deriving them from the base schema. This leaves it to the schema authors to maintain the coherence of all project specific document formats.

Many authoring tools support the editing of tables and lists only within an HTML editor. When using such tools as a technical basis we have to allow HTML mark up within the XML documents. As a consequence it will not be possible to assign metadata and alternative recourses to media elements that are edited within these HTML regions of the document. The management of the resource files of these 'HTML media' is an additional problem demanding for additional programming or user action.

2.5 Shareable Content Description Language (SCDL)

Within the abstract base format we define a small set of elements. For the three levels of our content hierarchy we define the following elements:

- *Sharable-content*: This is the base element for the conceptual document level. It may contain *sharable-content* and *content-block* elements, *metadata* elements and *relation* elements for cross references of child content units.
- *Content-block*: This is the base element for the structural document level. It may contain *content-block* elements, *table*, *list*, *media* and *parameter-list* elements. For tables and lists we provide elements with the same names as used within XHTML. Table cell and list entry elements may contain *table*, *list* and *media* elements.
- *Media*: This is the base element for the media document level. It may contain *text-valu*e elements, *resource* elements to reference a resource file or contain resource text like MathML, link *target* and *metadata* elements.

Starting from these base elements any authoring project has to derive its own set of content units as restrictions of the base elements. We developed a document format for one of the content projects ("Neue Statistik") managed by CeDiS.

3 Editor

Even the best document format is useless, if there are no appropriate tools to edit, manage and publish documents coded in this format. While searching for authoring tools, which can work with our document format SCDL, we first looked at classical XML editors (*XMetaL*, *XMLSpy*). These tools can work with our SCDL format as native document format and provide validation against the SCDL schema. Anyway, out of the box these XML editors are not comfortable authoring software, especially for a complex XML schema like SCDL. It is possible to create a convincing user interface on top of the XML editors, but only with heavy development effort. Another problem is cost for the end users. If we want to offer an authoring tool for broad use within the FU Berlin and maybe within other universities, it is important that the authors are not scared by extensive license fees.
We decided to use *Microsoft InfoPath* as the starting point for our development. *InfoPath* is new software which is originally intended for creation and maintenance of forms, where the form input is stored in XML. Being completely XML based *InfoPath* allows immediate validation of the edited content against an XML schema. As *InfoPath* offers a programming interface and a GUI editor it is possible to create a suitable user interface with considerable development efforts. Since Version 2003, *InfoPath* is part of *Microsoft Office Professional* and therefore available to many authors within the university at almost no extra cost.
Of course there are several authoring tools on the market for the creation of e-learning contents, which have a proprietary, non-XML, document format and which create HTML output. These tools are useful and adequate for projects with few authors and/or a small volume of content, but they are not subject matter of this paper.

3.1 Specification

An editor is supposed to be easy to understand and comfortable to use for authors with different backgrounds. To hide the XML nature of the documents from the author it needs a clear, intuitive and ergonomic graphical user-interface with the look & feel of a word processor. This is a rough list of features:

- To keep authoring comfortable, the editor should have common functionalities like undo & redo, cut & paste, spell checking and support for editing lists, tables and links.
- In an academic environment, non-ASCII characters from French to Arabic are in common use, therefore UTF-8 character encoding is absolutely necessary.
- Since e-learning projects differ widely in their requirements on how to structure their contents, the tool should be readily adaptable to diverse document structures.

- Mathematical formulas are a big topic in scientific e-learning-contents. Therefore the integration of a graphical formula editor is important.
- To enrich the content with multimedia components, the editor must support the common file formats for images, video, sound and animations.
- To give the author a good understanding of one of the final content representations, an HTML-preview is indispensable.

It is not convincing to offer a WYSIWYG authoring environment if the contents are supposed to be transformed to very different layouts and designs. We prefer a 'semi-WYSIWYG' solution that offers a uniform structural view of the document, with only having a few formatting options (e.g. bold, italic, colors and tables) accessible in WYSIWYG mode. Though, as an experience from former projects we have learned, that it is very important for the user's acceptance of the editor, to provide a project specific HTML preview.

3.2 Learning Module Editor LME 2005

The most evident aspect when choosing a commercial editor is a platform and vendor dependency. In case of InfoPath we obliged ourselves into the .NET—and Jscript-world, and the MS Office XML-Environment. But the produced XML is standardized and can be handled by any XML tool on the market. InfoPath keeps the content and the inserted data real-time validated against a SCDL Schema. Nevertheless, considerable development effort is necessary to achieve an editor—called LME 2005, which matches our requirements starting from the build-in features of InfoPath.
As InfoPath is part of Office Professional, it provides a working environment that is common for most of the authors at FU Berlin. But not only the usability and the well-known office ergonomics is a positive effect we obtain from the InfoPath solution. It is also very convenient that the same spell-checking engine and the same dictionary (including user specific extensions) are applied like in the other office applications. Furthermore, being part of Office Professional, InfoPath is available for many authors at the university at no or few extra cost.
Because InfoPath uses common Microsoft components to display an HTML based GUI, it can display media elements like flash, video and audio within the authoring GUI using ActiveX-controls. However, for mathematical formulas such a control is leaking. This is one of the missing features with the highest priority for future development. As a workaround formulas can be embedded as images like any other illustration. Nevertheless, for mathematical and technical documents, which contain many formulas, this is not reasonable.
Another constraint we had to accept was to declare all text paragraphs as XHMTL-fields. This was necessary as it was the only convenient way InfoPath supports to import and edit tables. LME 2005 has now the capabilities of Word-alike text formatting, including tables, under concession of bringing together style and content information, and loosing the possibility of assigning metadata and alternative resources to all embedded images in these paragraphs.
InfoPath provides a good foundation for the development of an editor, which meets our authoring requirements and technical specifications with only few compromises; but there are still some features missing like mathematical formulas. Also an adaptable user interface is needed for different document formats derived for different projects from the general SCDL document format. This cannot be done in InfoPath and must be developed as an additional application.

4 Outlook

An accepted standard for a document format for e-learning contents and a corresponding XML schema would be of great advantage both for the development of software for creation and management of content and for the exchange of content among different teachers. We hope that our document format description can be a valuable contribution to this standardization process.
Next steps in the development of our authoring environment would be the implementation of missing features like the integration of a formula editor and an editor for cross references. Further important steps towards a general applicability of our editor would be the development of a graphical editor for didactic structures and the automatic adaptation of the authoring user interface to the various document structures.
We suppose that many other universities are confronted also with the task to provide authors of e-learning contents with convincing solutions. Thereby we expect XML based solutions to play an important role, especially if sustainability and single source publishing are important issues. In the field of authoring of e-learning contents (XML based and non XML based) we are still looking for cooperation partners from other universities and from commercial companies.

5 Acknowledgements

We highly appreciate the support given to us by Microsoft Germany. It contributed much to the success of our efforts.

References

ADL Technical Team (2004). Sharable Content Object Reference Model (SCORM) Content Aggregation Model (CAM) Version 1.3.1, from www.adlnet.org.

CeDiS, from http://www.cedis.fu-berlin.de, 15.4.2005

DeRose, S., Maler, E., Orchard, D. (2001), XML Linking Language (XLink) Version 1.0, from www.w3.org/TR/xlink/, 15.4.2005

DITA TC 1.1 issues list. from www.oasis-open.org/committees/download.php/12120/DITA-TC-1dot1issues-1.htm, 15.4.2005

Juhnke, N., Herrmann, N. (2004), e-Learning-Manufaktur – Ein Produktionsprozess für flexibel einsetzbare e-Learning Materialien, in: K. Rebensburg, (Ed.), Grundfragen Multimedialen Lehrens und Lernens (pp. 229-240). 2. Workshop GML 2004

Learning Net, from www.cedis.fu-berlin.de/plain.php?cont=1256, 15.4.2005

LMML, from www.lmml.de, 15.4.2005

<ML>3—Multidimensional LearningObjects and Modular Lectures Markup Language, from www.ml-3.org, 15.4.2005

Neue Statistik—Willkommen auf unseren Seiten, from http://www.neuestatistik.de/, 15.4.2005

New Economy Projektseiten, from http://www.internetoekonomie.com/, 15.4.2005

Priestley, M., Hackos, J. (2005), OASIS DITA Architectural Specification Committee Draft 01, from http://www.oasis-open.org/committees/dita/, 15.4.2005

Sperberg-McQueen, C. M., Burnard, L. (eds.) (2002). TEI P4: Guidelines for Electronic Text Encoding and Interchange. Text Encoding Initiative Consortium. XML Version: Oxford, Providence, Charlottesville, Bergen

Walsh, N. (2002), The DocBook Document Type Committee Specification 4.2, from www.oasis-open.org/docbook/specs, 15.4.2005

XLML, from www.xlml.org, 15.4.2005