Modules for an XML Schema in the book-on-demand process

Klaus Kreulich
Chemnitz Technical University, Institute for Print and Media Technology, Chemnitz, Germany
klaus.kreulich@mbv.tu-chemnitz.de
http://www.tu-chemnitz.de/pm
Arved C. Hübler Prof. Dr.
Chemnitz Technical University, Institute for Print and Media Technology, Chemnitz, Germany
arved.huebler@mbv.tu-chemnitz.de
http://www.tu-chemnitz.de/pm

Abstract:

In this paper we discuss an XML Schema design approach for developing Book-on-Demand applications. We consider the Book-on-Demand (BoD) process and discuss an implementation of it as an XML application. Our starting point is an analysis of the BoD process. As evaluation result of today's BoD technology, we propose an up to date BoD workflow model. The presented model enables the user to produce his or her personal book out of a structured information base, which is for instance provided by a publisher website. Within the production workflow, the user is allowed to select parts of relevant documents, to structure them by his own purpose, to determine a layout of his choice and forward the resulting new document to a Print-on-Demand Provider.

We review an XML-based approach for technical implementation of the presented BoD workflow model. With regard to the importance of the PDF format for the publishing and printing industry, we include a short discussion about an alternative approach which is based on an underlying PDF document information base.

The XML approach requires an appropriate XML Document Type Definition (DTD) or in view of more automatic processing an XML Schema. XML Schema provides new opportunities for Book-on-Demand applications, in particular it directly supports subtler content management methods for individual user requirements. The paper illustrates how to use and implement these capabilities.

Introduction

Presently, XML applications are spreading into all fields of electronic data processing. XML, the simplified variant of SGML, is perfectly suited for data exchange formats, as well as for data container formats. As simplified SGML it offers all fundamental advantages of its complex predecessor without losing essential functionality. Especially, the success of XML drives on further improvements and developments of the standards. For the time being, one of the most important renewals is the development of an XML Schema Specification . XML Schema eliminates substantial shortcomings of the Document Type Definition (DTD) concept. This way the basis for the next generation of automated Web applications is established.

The publishing and printing industries profit from this development, as well. In these industries, XML can be used within the scope of Book-on-Demand (BoD) applications. In typical BoD applications data units are compiled on-the-fly according to a user request and printed as a book. BoDs can be characterized as products of a database production process. Within this process the user is provided with deeply structured data through an Web interface. This is exactly where the power of XML lies in. Therefore, an XML based modeling of BoD applications is self-evident.

Book-on-demand applications

In parallel to the development of digital printing and the progress in Print-on-Demand (PoD), new concepts regarding the BoD production are continually evolving. The varied applications of these concepts can be divided into different groups according to their objectives:

Limited Edition Books The PoD procedure complements the mass production of books for such occasions that short run jobs cannot be realized economically. Typical examples are re-prints, rare books, or specific technical books. Thus the books being printed in principle do not differ from conventionally printed books but only from their print volume.

Conditioned Books BoD applications are used to produce books or booklets with so-called conditioned document parts. That is, subject to relevant boundary conditions different versions of the documents can be printed. One of the most frequent uses of this BoD procedure is the production of technical manuals. An example for this is the instruction manual of a car. In that case the manuals of one model line are produced on the basis of one printing copy, whereas the specific features of the respective models, such as the motorization, are not fixed before the print start.

Individual Section Books This concept allows the user to select large corporate structure units, e.g. individual stories, reports or recipes, from an existing database via web browser and to compile a new book with. The arising book may contain an adjusted table of contents and uses personal information of the user, e.g. his name, to personalize the book. This BoD application is a consistent extension of the limited edition application that combines the options of the PoD technology and those of database and Internet technology.

The current approaches are only a first step towards a publication workflow that is completely brought into line with the user's demand. A consistent extension of the existing BoD procedures has been discussed for several years, see e.g. , . The objective is the inclusion of the user into the editorial process. The user is to get more options to edit the publication copy. Simultaneously, an automation of this production stage is possible. Concrete demands on the next BoD generation, in the following also called "Generic Books", are:

Open Structures The flexibility of the selection opportunities concerning the book content must go beyond the fixed bounds of logic structure units like chapters or sections. An exchange of element like tables, images, or paragraphs among different chapters and also different books should be possible.

Third-Party Data Integration Beside the data offered directly on publisher websites or by a content-provider, the integration of data from other sources should be possible. Examples for this are user documents and images or contents of other providers. Thus the user can get contents from different publishers or content providers and can join them together to a new book.

Automatic Semantic Compilation The user is given the option to indicate a profile of the contents for his book. According to a database that is deeply structured and provided with suitable meta data a software compiles a book proposal.

Flexible Layout Opportunity During the production the layout of an individual publication should be adjustable according to user needs and/or publication contents.

Automatic Book Structure Enhancement On user request the BoD application should generate book typical indices like a glossary, an index or a table of content. The enhancement can be drawn up on the basis of a semantic tagging of single document parts, on meta data, on the logical book structure, etc.

At present there are some BoD approaches which are on the way to Generic Books. An example is the European Union founded project **TRIAL-Solution** . The objective of this project is to develop mathematics textbooks which can be composed on an individual learning level. Another example is the project **Study-Guide-on-Demand** at Chemnitz Technical University. In this project study information can be put together to an individual booklet and offered via the Internet.

Book-on-demand production

Production chain

At the production of a conventional book the reader or user stands at the end of a linear process chain. The author compiles book contents in a manuscript while the publisher is in charge of the book's presentation and marketing. In the pressroom the book is printed and, finally, the book is offered at the bookstore.

The process chain for the production of a BoD is more open to the user. The user accompanies the production of the book from selection of the contents over layout to printing. With the help of a future BoD system the users could generate the books by themselves and subsequently enlist the service of a PoD provider. In practice for this, however, the problem of copyright has to be settled.

In respect of the technical sequences the BoD production chain can be subdivided into five fundamental stages:

1. Selection Relevant information or contents are selected in a book. The selection process can be done by the author, a publisher or content provider in general, and by the reader. Subject to the database a completely automated software process is possible, as well.
2. Structuring The selected book contents are sequenced logically. In case of a just-in-time production this process stage should be done in direct communication with the reader or automatically.
3. Formatting A suitable layout and book options regarding material and size are to be fixed. Within this process stage the composition dependent references, such as page numbers in the table of contents or footnotes, are formatted. The formatting as well as the structuring should be done automatically or in dialogue with the reader. The result is a finished printable ontents can be done independently from later decisions concerning the layout. The structuring model of XML supports the decomposition of books into user relevant semantic units. Therewith the basis for a free combination and compilation of document modules into individual books is given. Generally, the defined logic structures of XML documents simplify an automatic information processing. Besides the processing of textual data, the BoD production requires the processing of data for the control of the entire workflow and data for the course of business. These are typically data from relational databases for whose processing XML is perfectly qualified. As reference for the use of XML in these fields all current XML based developments in the E-Business sector can be seen. Within the bounds of this description the control and business processes are not to be deepened. In principle, a complete workflow with XML data from selection to printing and post-press processing would be possible, in practice XML, however, is not ripe for supporting the production of professional print products. The lacking prerequisites to ensure a continuous XML data flow are suitable formatters for the output devices. In this field the current practice shows that, for the next years at least, PDF will be the dominant data or document format for the printing industry. For the time being, PDF processing is replacing the PostScript based production workflow in the printing plants. In the course of this the fundamental problems of PostScript documents, the high error rate and the immense rendering times, are eliminated. According to it being device-independent, PDF itself does not contain control data. For these purposes several print workflow data formats were developed, so, among others, Adobe defined the Portable Job Ticket Format (PJTF). With the help of PJTF even job data and post-processing data are transported. Therefore, PDF and PJTF form an ideal basis for an automated print production workflow as demanded by the BoD production. Subject to the product objective three suggestive options for a BoD data-workflow model can be deduced from the general considerations concerning XML and PDF:

4. XML / PDF data-workflow Firstly, the data are modeled into XML and during formatting the PDF documents are generated, in which the necessary print-workflow data are embedded as PJTF data. According to the state-of-the-art technology, on the basis of this Generic Books with any flexibility and high printing quality at the same time can be realized.
5. PDF data-workflow A workflow that is completely based on PDF documents or PDF book modules offers a convenient solution from the technical point of view. However, this variant only offers the option of a "conventional" BoD application. PDF is primarily suitable as continuous data format when a pre-formatting is desired and the resulting disadvantages (see above) regarding a flexible book structure are not relevant. As page description language, PDF offers exact and broad information concerning the layout of individual pages. Individual objects like images or charts are exchangeable. With the help of qualified tools, simple changes of the text and the exchange of individual symbols is possible. However, yet a change that includes a wordwrap is unrealizable. Another big problem is the precise addressing of document parts. An exact mapping to a certain passage, as it is given by the document tree in XML, does not exist. Therefore the precise exchange of document parts is laborious. On this point, neither the latest development library, the Adobe "PDF Library" carries a fundamental change, although the concept or function "Placed PDF" takes a further step towards a flexible editing possibility. Placed PDF can convert individual PDF pages into Encapsulated Postscript (EPS) objects, but these objects can only be embedded in consisting PDF documents as whole pages.
6. XML data-workflow As explained above, this variant is only possible if in future the direct control of printing and finishing machines via XML is possible. In this case, this variant will offer a consistent solution for Generic Book applications.

Irrespective of up to which process stage in BoD data-workflow is operated with XML data, for the use of XML a qualified data modeling is to be developed. The current method for realizing XML data models or XML applications is the definition of Document Type Definition (DTDs).

Book-oriented DTDs

From the many uses of XML or SGML in the field of automated document management many DTDs have evolved since the passing of the SGML standard in 1986. In the meantime, some of these DTDs have established themselves as official or unofficial standards. Some prominent book oriented DTDs are:

ISO 12083 DTD The publishing-industry DTD for books, serials, and articles.

DocBook DTD The computer-industry DTD for technical documentation.

Text Encoding Initiative (TEI) DTD A DTD used for literary and other research material.

MIL-STD-38784 DTD A U.S. military DTD from the CALS initiative, used for technical manuals.

HTML and XHTML The DTDs used for publication over the World Wide Web.

OEB DTD A DTD for presentation of E-Book content.

In the development of BoD applications these DTDs can be used as the basis for modeling textual structures. Thereby, however, the universal weaknesses of DTDs have to be considered.

A serious disadvantage of DTDs is that they only offer very inexact data types. At the end of a hierarchic element definition stands a `#PCDATA` element. This means that the content of the element can consist of any number of symbols. Additionally, the attributes can only weekly be typed in XML. In practice this means

that data often cannot be checked, as regards correct contents, by the standard means of an XML parser. For example a typical DTD definition for a date can be:

```
<!Element date     (year,month,day)>
<!Element year     (#PCDATA)>
<!Element month    (#PCDATA)>
<!Element day      (#PCDATA)>
```

According to this definition for an XML parser the following date is valid:

```
<date>
    <year>this is not a year</year>
    <month>04</month>
    <day>10</day>
<date>
```

In an automatic finishing process, especially in the communication between different business partners, such as a PoD provider and a publisher, expendable measures of protection have to be made in order to extract the meant digits. In other words, in the case of DTDs the burden of adding program logic to deal with unspecified data falls on the developer.

Another disadvantage of DTDs is the insufficient support to reuse content models. DTDs offer the mechanism of the parameter entities, that is a comfortable way to use strings repeatedly. However, a logic link between two elements with different names but similar content models is not possible. The above listed DTDs provide some subtle solutions for customizing them in particular application relevant contexts. However, all these solutions are based on entities, wherefore processing programs generally have to be adjusted to the customized elements. With the assistance of the type concept, XML Schemas offer new opportunities.

As indicated above, the completion of a production process and the realization of a business are parts of the BoD workflow. Data concerning the ordering process and the print and postpress workflows are to be taken under consideration. In this context, the DTDs above are to be complemented by suitable DTDs of other usages or newly developed DTDs. For a versatile application, a modularization of the entire BoD DTDs into a content share, a control share, a printing share, a postpress-workflow share etc. would be useful. The existing mechanisms for modularizing DTDs are rudimentary. XML Schemas mean improvements in this area, as well.

Benefits of XML Schema

To overcome the shortcomings of DTDs the W3C chartered a new Working Group that is concerned with the development of an XML Schema Recommendation. The current working draft contains fundamental approaches to improve a data management with XML. Overall XML Schema simplifies the automatic processing of data and documents. The interface between XML documents and databases is continually simplified and so the handling of dynamic data is supported.

One of the fundamental improvements compared to DTDs is a uniform syntax for document instances and document definitions; XML Schemas are XML Instances of the W3C Schema definition themselves. The processing of XML Schemas and XML Instances by the same software tools is possible. An automated processing of XML Schemas can be done on a much higher level than that of DTDs. In addition, less expenditure for the data management is necessary. The administration of the data as well as the programming expenditure is eased.

An important extension of the existing DTD properties is the opportunity to use strong typed data. XML Schema offers, similar to a modern computer language, an extensive quantity of implicit data types. Beyond it, within XML Schemas deduced or completely redefined types can be declared.

A further important property of XML Schemas is the support of **Namespaces** , with which the reuse of entire XML Vocabularies and also individual structure definitions is made easier. The Namespace concept was developed independently from XML Schemas, but in connection with XML Schemas for the first time a workable option for use is possible.

The reuse of Schemas or parts of them is supported by an object oriented approach in type definition. Like the categories of an object oriented computer language can inherit their properties from a basis category, the transition of properties between hierarchic types of XML Schemas is facilitated. The so-called deduced types can have exactly the same properties as their basis types, but they can also be modified by restrictions or extensions.

From the novelties and improvements that XML Schema offers various uses of BoD applications can be deduced. The following examples show several opportunities:

Example 1: Use of data types

Data types can be used in BoD applications in many ways. A typical example for the field of meta data is the ISBN of a book. While within a DTD element specification the ISBN is to be defined as an unspecified sequence of symbols, a Schema arrangement can represent an exact specification:

```
<element name="isbn">
    <simpleType base="string">
        <pattern value="\d{1,3}-\d{1,5}-\d{1,5}-(\d{1}|[A-Z])" />
    </simpleType>
</element>
```

According to this definition, an ISBN number is a character string of 1 to 3 digits, followed by a hyphen, followed by 1to 5 digits, followed by a hyphen, followed by 1 to 5 digits, followed by a hyphen, followed by either a digit or a capital letter. A respective and valid part of XML instances would be:

```
<isbn> 1-456-4897-X </isbn>
```

Example 2: Use of occurrence attributes

In DTDs the frequency of an element within a structure can be exactly one or can be fixed for one of the possibilities "zero or one", "one or more" or "zero or more", with the help of the occurrence indicators "?,+,*". XML Schemas provide the attributes "minOccurs" and "maxOccurs", with which an exact number of repetitions can be defined. An example for the optimization of BoD applications are details on limitations of the book pagination, such as limiting the number of chapters or limiting the images within one passage etc.

```
<element name="chapter" minOccurs="3" maxOccurs="10">
```

The definition signifies that the element "chapter" can occur exactly 3 to 10 times. Such a limitation can be particularly useful in connection with a respective cost model for a BoD.

Example 3: Use of inheritance

A typical inheritance application is the connection between a publisher's series and a particular book of this series. So, a publishing house could define a universal type for academic publications within an XML Schema and could deduce the characteristics of informatics books from that.

Example 4: Use of Namespaces

The Namespace concept enables other communities to quite simply use the Schemas. Besides the reuse of existing Schemas, a modularization of a BoD Schema is supported, as well.

Assuming that in future the communities involved in the BoD process will define appropriate Namespaces and belonging XML Schemas, a universal XML Schema could use all these Schemas directly, with the help of the "import mechanism". A part of a universal BoD Schema, i.e. based on Dublin Core meta data, the Adobe JobTicket format, and the ISO 12083 book model could be as follows:

```
<schema xmlns="http://www.w3.org/1999/XMLSchema">
...
    <import namespace="http://www.purl.org/DC#"
            schemaLocation="http://www.purl.org/DC/dc.xsd"
    />
    <import namespace="http://www.adobe.com/PJTF"
            schemaLocation=""http://www.adobe.com/PJTF/pjtf.xsd"
    />
    <import namespace="http://www.iso.org/../iso12083"
            schemaLocation="http://www.iso.org/../iso12083/iso12083.xsd
    />
...
</schema>
```

Conclusion

Future BoD applications will mean further progress regarding individualization and personalization of books. This progress will affect the structural composition as well as the layout of the products.

The technical realization of BoD can be divided into two production sections. The compilation of the contents on the website of the supplier and the book production by the PoD provider. XML is the predestinated format for the section. Presently, PDF is about to establish itself within the second section and to replace PostScript workflows. Here, XML is (still) secondary.

The use of XML Schema instead of DTDs promises a considerable automation of the processes. However, firstly, XML Schemas have to win recognition as a new standard against the widespread and established DTD-based applications. Despite all described advantages, it remains to be seen how quickly high-performance XML Schema parser and respective tools will be available.

Bibliography

[Schema 00] World Wide Web Consortium: XML Schema Working Draft, Feb 2000. **http://www.w3.org/TR/xmlschema-1/**

[Ahon 96] H. Ahonen, B. Heikkinen, O. Heinonen, J. Jaakkola, P. Kilpeläinen, G. Linden, and H. Mannila. Intelligent assembly of structured documents. Report C-1996-40, Department of Computer Science, University of Helsinki, 1996.

[Kreu 98] K. Kreulich: The Generic Book as an Application of Intelligent Information Retrieval Systems, Abstracts of the 22nd Annual Conference of the German Society for Classification, Dresden, March 1998, S.65

[Trial] Interoperability and Intelligent Reuse of Distributed Teaching Materials. The TRIAL-Solution Project. **http://www.trial-solution.de/**

[SGoD] Study-Guide-on-Demand Project of the Institute for Print- and Media Technology at Chemnitz Technical University, Germany. **http://www.pm.tu-chemnitz.de/sf/**

[NameSp 99] World Wide Web Consortium: Namespaces in XML Recommendation, Jan 1999. **http://www.w3.org/TR/1999/REC-xml-names-19990114/**

Authors

Klaus Kreulich

Academic
Chemnitz Technical University, Institute for Print and Media Technology
Postal Address:
Reichenhainer Str. 70
09126 Chemnitz
Germany
Telephon: +49-371-531-8091
Fax: +49-371-531-3780
E-mail: klaus.kreulich@mbv.tu-chemnitz.de
Web: www.tu-chemnitz.de/pm

Klaus Kreulich - Klaus has been working as academic at the Institute for Print- and Media Technology at Chemnitz Technical University since 1997. He is involved in various research projects concerning XML and Print-on-Demand technologies. At present Klaus is working on his Ph.D. Thesis concerning XML publishing concepts.

Arved C. Hübler Prof. Dr.

Director of Institute
Chemnitz Technical University, Institute for Print and Media Technology
Postal Address:
Reichenhainer Str. 70
09126 Chemnitz
Germany
Telephon: +49-371-531-8091
Fax: +49-371-531-3780
E-mail: arved.huebler@mbv.tu-chemnitz.de
Web: www.tu-chemnitz.de/pm

Arved Hübler - Prof. Dr. Arved C. Hübler, Director of the Institute for Print and Media Technology [ ] at Chemnitz Technical University, formerly was Technical Director with the Bertelsmann Group in Gütersloh, Germany. At pm Institute, several projects in PoD technology relating to PoD workflow, new book binding techniques, XML document structuring, automated book generation and document digitalisation are in progress. In addition consulting projects in implementing PoD production in several companies where done. Arved Hübler is Member of the TAGA. He joined several TAGA, IARIGAI, IS&T and other conferences.