Value-Added Digital Libraries Service: Individualized Crossmedia Output with XML Documents

Klaus Kreulich
Scientist, Institute for Print and Media Technology
Technical University of Chemnitz, Germany
Email: klaus.kreulich@mbv.tu-chemnitz.de

Eckhard Stasch
Scientist, Institute for Print and Media Technology
Technical University of Chemnitz, Germany
Email: eckhard.stasch@mbv.tu-chemnitz.de

Arved C. Hübler
Professor, Institute for Print and Media Technology
Technical University of Chemnitz, Germany
Email: arved.huebler@mbv.tu-chemnitz.de

Abstract:
This paper introduces a concept for creating documents, which are gathered, annotated, and structured individually, on the basis of a common digital document collection. Possible output media are Print-on-Demand booklets as well as PDF documents or E-Books. A prototypic implementation of this concept, an XML based online study guide is described. The paper focuses on the particularities of the underlying XML Document Type Definition and the technical realization of the study guide.

1. Introduction

The rise of the Internet has opened a new world of possibilities for the digital exchange of information which led and still is leading to deep structural and conceptual changes in the library sector. The classical printed book loses its position as the most important information carrier and is strongly challenged by online information resources. One of the most striking features of digital online information is the fast and flexible access to the body of human knowledge which is spread throughout the world. This flexible accessibility derives from the fact that digital information can be processed automatically which helps to overcome what one of the greats of library science, J. R. C. Licklider [1], used to call *the passiveness of the printed page*.

But the advantages of digital information resources are opposed by some fundamental disadvantages and shortcomings still waiting to be solved, balanced, or negated. First to mention is the presentation of digital information, which is most commonly displayed on a computer screen. Depending on its scope and content, information is only partially suited to be read on a monitor. Especially great amounts of information which ask for deeper and comprehensive reading are utilized more comfortably and effectively in printed form [2]. Another problem with digital information is that searching for specific content often floods the user with information that is not relevant. The wider the range of an information basis the more crucial is the possibility to make specific selections - enabling not only a "macroscopic selection" among different documents but also a "microscopic selection" within one document.

One new trend in the search for usable information facilities is the demand for dual-form documents which combine the advantages both of electronic and paper versions. The request for printed versions of an individual selection of information arises in many applications. After having been selected by a user each specific sample of information should still be freely arrangeable and processable in order to create a structure and a layout which provides optimal usability.

A solution for this task can be based on the new standard XML [3]. This standard offers an easy way to structure documents with respect to their content and allows for providing different ways to present them. The user has the opportunity to select small information units suited to his current individual needs, compose these units to a new document, and display this document by an output medium of his choice. If this concept is combined with an appropriate Print-on-Demand (PoD) workflow, the user will be delivered a printed document that is full of relevant information and structured individually. In order to realize this and other XML-Publishing solutions the Project "Individualized Structured Study Information (ISSI)" has been launched at the Institute for Print and Media Technology in Chemnitz

Beside the ISSI project there are a lot of projects concerning individualized information access. A few examples with close relationship to ISSI are: 1. The Digital Library project at the Göttingen Digitization Center [4] in Germany. An XML format is applied to store digitized documents. A PoD access to document parts is planned. 2. Just at the beginning is the European Union funded project "Interoperability and Intelligent Reuse of Distributed Teaching Materials" [5]. One of its aims is to provide a Digital Library with structured teaching material for personal knowledge needs. 3. Another example is the project

"SGML/XML-based Thesis and Dissertations - Cooperations and Archives" [6]. It is part of the German and international activities concerning electronic dissertations.

The following sections of this article describes the conception and implementation of an online study guide within the ISSI project. Section 2 sketches the general concept and describes the technical structure of the study guide application. Section 3 outlines a general workflow for an on-demand-production of individual books. It is called a "Generic Book" workflow and is a consequent extension of the online study guide concept. Section 4 ends up with a final conclusion and prospects for further works.

1.1. XML Basics

Before we proceed with the ISSI concept, for those readers who are unfamiliar with the idea and syntax of XML, we give a short overview of the basics necessary for the understanding of this article.

XML is a meta-language for defining document structures. A concrete XML application is defined by a so called Document Type Definition (DTD), wherein the logical structure of the XML application documents is specified. An often used example is a DTD for encoding emails in XML:

```
<!-- simple email DTD -->
<!ELEMENT  email          (receiver, sender, subject?, body)>
<!ATTLIST    email
        importance     (high, medium, low) 'medium'>
<!ELEMENT receiver        (#PCDATA)>
<!ELEMENT sender          (#PCDATA)>
<!ELEMENT subject         (#PCDATA)>
<!ELEMENT body            (#PCDATA)>
```

The first line is a comment, the syntax *<!-- ... -->* is equal to the comment syntax in HTML documents. The following line in the DTD defines the element *email*. The declaration consists of two parts, the name of the element and the content model which is specifying what can appear within the element. According to the content model the possible structure of an *email* element is given as follows: the element *email* must contain the element *receiver*, followed by the element *sender*, followed by the optional element *subject*, and the last element within *email* has to be *body*. Complementary to element declarations attribute declarations are allowed also. In XML attributes provide meta-data for elements. In the example an *email* element can be of high, medium or low importance with the default value medium. The remaining four declarations are trivial, they allow character data only as content for the elements *receiver*, *sender*, *subject*, and *body*.

Within the XML instances or the XML documents, respectively, the DTD corresponding structure is marked up with "tags" written in left and right angle brackets. An example for a valid XML document according to the above DTD is:

```
<!-- xml document according to simple email DTD -->
<?xml version="1.0"?>
<!DOCTYPE email SYSTEM "email.dtd">
<email>
        <receiver> John Smith </receiver>
        <sender> Peter Smith </sender>
        <subject> Meeting </subject>
        <body> Hi John,
                the meeting starts at 10 am.
                Peter
        </body>
</email>
```

The document begins with a comment and two declaration lines. The content is included in the root element *email*. To process this document an XML parser or more generally an XML application is necessary. In general the application uses the DTD to identify the logical structure of the document. According to its inherent rules the application decides what is to do with the various document parts, e.g. searching in a local network for the person defined by the element *receiver* and let the content of the element *body* pop up on his monitor.

In the example no instruction for the presentation of the document is given. It is one of the strengths of XML to separate formatting from content. The application is able to use various style sheets to present the document on different output media with different formats.

2. Individualized Structured Study Information

2.1. Objectives and Concept

The general aim of the ISSI project is the development and realization of innovative publication models, which enable the automation of editing information according to individual user demands. Supported by an intelligent system a user is able to generate personal products from an extensive information basis. As the first prototype application of the described concept a study guide for the Institute for Print and Media Technology was realized [7].

This study guide is an XML application which provides the user with some documents for a personal selection of information. Within a standard web browser the user is able to gather document parts in a so called "print-basket". Furthermore, the user is offered some special data input fields which allow him to annotate the documents with personal remarks. The content of the print-basket and the annotations can be printed automatically and delivered to the user by mail. As an alternative to the printed documents, the user receives a corresponding PDF file with an automatic e-mail. To realize this functionality an adequate XML Document Type Definition (DTD) was developed and documents of the study guides were implemented as XML instances of this DTD.

In the first version of this study guide information about the subject 'print and media technology' is offered. In later versions information about all fields of the Technical University Chemnitz will be integrated gradually. With regard to the contents, the conception of the study guide corresponds to those of classic print products. The emphasis is put on typical study relevant themes such as lecture descriptions or field profiles.

In contrast to exclusively paperbased study guides the reader is given the possibility to select contents individually. The reader is able to find information online, to select relevant parts, to add personal remarks to the selected items and so to generate his personal study guide.

In principle, every piece of information is available for every interested person on the output medium of his choice. At the moment, a screen-version and a PoD paper-version are realized. An E-Book interface is further planned.

2.2. Document Type Definition

The functionality and flexibility of an XML application is mainly determined by the underlying Document Type Definition (DTD). In order to define a suitable and general document structure the contents as well as the possible usage of the documents are to be considered. In the case of the described study guide particularly, the structural variation possibilities to assemble an individual study guide are directly dependent on the DTD.

The contents of the study guide are basically text or book oriented, which means the logical structure of a study guide document can be characterized by typical book structures such as continuous text organized in paragraphs, sections, chapters, and so on. Correspondingly, the design of the underlying DTD was orientated towards established book oriented industrial DTDs.

Standard DTDs

One of the main matters of concern of SGML standards was the development of a generally valid data exchange format. According to the aim to exchange documents beyond the limits of application and platform in several branches of industry standardized DTDs were developed. Among the most widely spread book oriented DTDs are the following:

- ISO 12083 [8]
  The publishing-industry DTD for books, serials, and articles.

- DocBook [9]
  The computer-industry DTD for technical documentation.

- Text Encoding Initiative (TEI-Lite) DTD [10]
  A DTD used for literary and other research material.

- MIL-STD-38784 [11]
  A U.S. military DTD from the CALS initiative, used for technical manuals.

- HTML [12]
  The DTD used for publication over the World Wide Web.

The study guide DTD contains a few concepts and ideas of these DTDs and in particular of the ISO 12083 standard. Special additions were made mainly according to the specific demands of the study guide for structural variability and the realization of a user input interface.

Structural Variability

One of the principal objectives within the ISSI project is the realization of individual possibilities to print and select single document parts. However, to arrange the document parts in an optimal or user specific overall structure a high variability with regard to valid structures is necessary. In view of these demands, for defining the content-models of the study guide DTD the following simple heuristic rules are used:

- Prefer choice operator "|" against sequence operator ","

- Prefer optional occurrence indicators "*,?" against required occurrence indicator "+"

By a consequent use of the choice operator instead of the sequence operator "," obviously a higher flexibility of the logical structure within the element content-models is attained.

The use of so-called "dummy documents" is made possible by the second rule. A typical problem with processing document fragments arises when they are evaluated as non valid with respect to the DTD - because of missing elements. A solution is to process the required, missing elements without their contents. A simple realization within the element definitions is obviously the use of optional contents [13].

For example, applying the above-mentioned rules to the study guide element *lecture* means:

```
<!ELEMENT  lecture
        (Title?, (lecturer* | abstract? | keyword_list? |
          conditions? | exercises? | literature? | other?)*)>
```

All elements of the content model are optional, and with respect to order only the title of a lecture is fixed as the first element. This leads to the desired variability. Disadvantages only appear with the other possible uses of the study guide documents. A plain processing within a parser of a third party XML application is in general achieved by a more prescriptive DTD [13]. For this the following definition would be more useful:

```
<!ELEMENT lecture
        (Title, lecturer*, abstract?, keyword_list?,
          conditions?, exercises?, literature?, other?)>
```

For the actual selection possibility of single document parts or DTD elements the attribute *select_id* and the attribute *order* are used. All possibly eligible elements receive both these attributes. The list of attributes for the element *lecture* is for example:

```
<!ATTLIST lecture
        select_id       ID              #IMPLIED
        order           (yes | no)      "no">
```

The attribute *select_id* serves, according to its type, as unique identifier of the elements. The default value is set to #IMPLIED within the DTD so that it is not fixed which elements are intended as an individual selection. When creating a new document instance the editor decides, by indicating or leaving away the attribute *select_id*, whether a single element is available for an individual selection or not.

To every *select_id* attribute belongs an *order* attribute. This attribute is used as the return value of the user's selection process. The *order* attribute indicates to the XML parser if the corresponding element was chosen or not.

User Input Interface

The individuality of the study guide is realized, among others by the opportunity of personal user inputs. For these inputs the DTD provides three different elements:

1. user_name

```
<!ELEMENT  user_name      (#PCDATA)>
```

The element *user_name* is used for the personalization of the study guide. That is why it is explicitly contained in the content model of the root element *study_guide*:

```
<!ELEMENT  study_guide    (title,subtitle?,user_name?,...)>
```

This element structure makes it possible to create a cover or any other introductory part with the user’s name on it. The element *user_name* is furthermore used in the content model of the element *para*. As indicated by its name the element *para* stands for paragraphs within the hierarchy of the study guide DTD. Established on this level the user name can be used virtually in the entire document.

The print-version of the study guide also displays the user name in the footer of each page. This function is not based on a special DTD element. Rather it is enabled by a feature of the used desktop publishing software FrameMaker+SGML[1] (see below) which allows the positioning of certain elements within the footers while formatting the XML documents.

2. user_annotation

```
<!ELEMENT  user_annotation (title?, content?)>
<!ATTLIST    user_annotation
             annotation_ref          IDREF           #REQUIRED>
```

This element is used to provide an interface for personal annotations at the user’s disposal. The annotations on a selected paragraph that the user enters online can appear independently from this paragraph at various locations of the study guide. For example a list at the end of

[1] FrameMaker+SGML is a trademark of Adobe Systems Incorporated.

the entire document is possible. The reference to the annotated document part is realized by the attribute *annotation_ref* . This attribute refers to the *select_id* attribute of the corresponding document part.

3. user_slogan

<!ELEMENT user_slogan (#PCDATA)>

This element was introduced to emphasize the individual character of the study guide. The element *user_slogan* allows the user to import any sentence into the entire study guide, i.e. his personal study slogan.

The elements *user_name* and *user_slogan* do not differ from each other syntactically. *User_slogan* is also integrated in the content model of the element *para*.

2.3. Technical Architecture

The realization of the study guide sets into action several soft and hardware components. The hardware for PoD processing and the computer technology consist of standard components and will not be described in this paper. For the software components standard products are adapted to the study guide requirements as far as possible. Some other additional modules are developed from scratch. A scheme of the technical architecture is shown in figure 1.

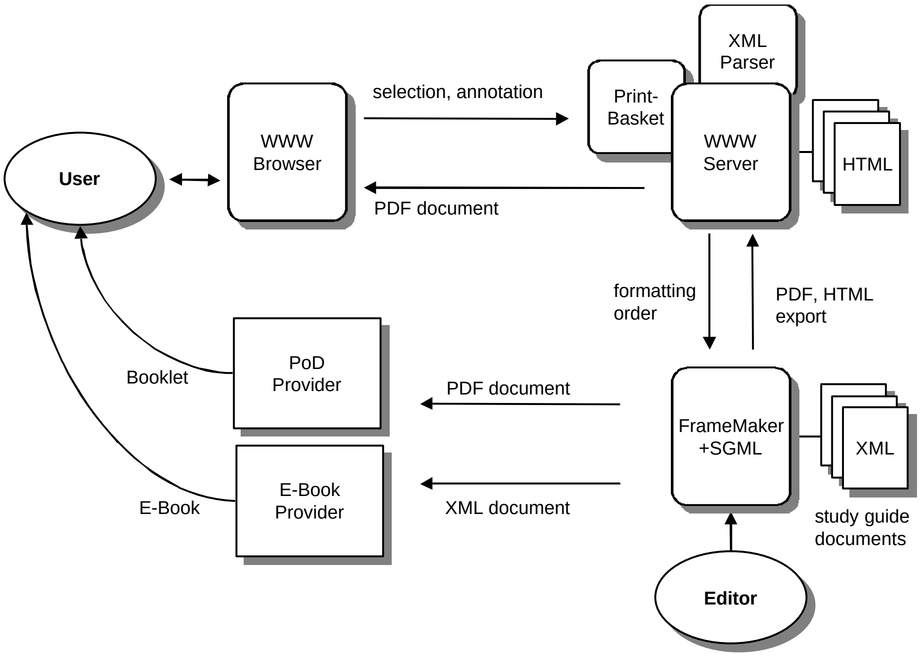

Figure 1: Technical Architecture

As user-interface serves a standard web browser which has to be capable of JavaScript, i.e. Netscape Communicator or Microsoft Internet Explorer. The capability of JavaScript is necessary demand for the print-basket concept described below. The requests from the web browser are responded to by a web server. Thus, the most important technical demand to the web server is a CGI-interface for supporting different Perl scripts.

The XML-parser software is grounding on the Perl module "XML:Parser" [14]. The realization of the PoD concept and other essential adjustments are implemented within the calling Perl script.

For the formatting and editing of the XML documents FrameMaker+SGML and several Perl scripts are used. Above all, FrameMaker+SGML is used because of its ability to format XML documents as PDF documents. The PDF format is used in the ISSI Project for the PoD process as well as for the screen-version of an individual document.

In the following sections two important processing workflows are described: first that of a user requirement and second that during the editing of a study guide document.

2.4. Request Workflow

A user of the study guide is given the opportunity to compose and create an individual document. The foregoing selection of document fragments and the order is done through the internet. The medium for delivering the paper-based study guide is traditional mail. The whole process can be described as follows:

1. Content selection and annotation
   The user selects document fragments which seem relevant to him and drops them into the print-basket described below. Each of these fragments can be annotated with personal remarks.
2. Order
   The user fills in an HTML form and sends the order to the web server via HTTP. The order consists of originator information and the user's document choice.
3. Reception of the order
   With receiving the order the web server creates a "job-ticket" which traces the present status of the order processing. Besides that a confirmation of the reception is send to the order originator.
4. Structuring
   In the next step the selected document fragments are assembled to a complete study guide. Therefore, an XML parser is used. The parser takes the document fragment order list and forms a valid XML document according to the study guide DTD. Several techniques for structuring the resulting document are planed. In the current prototype the above-mentioned dummy document procedure is implemented. This means that the logical structure of the original study guide is preserved. In later versions algorithms will be implemented which can generate a new overall structure based on the structural variability of the DTD.
5. Formatting
   The resulting XML document is forwarded to FrameMaker+SGML. By means of the DTD and the formatting rules a PDF document is created.
6. Printing
   If the user requests a print-version, the PDF document is processed in a PoD workflow. It is automatically printed and bound to a booklet.

7. Delivery
   The individual study guide is available for the user as crossmedia product. The delivery depends on the medium. The following possibilities can be used:

- Paper
  The print-version of the study guide is produced on-demand and is delivered to the user via conventional mail.
- PDF document
  A PDF document is sent by automatic e-mail. For reading on screen the free Adobe Acrobat Reader[2] is available.
- XML document
  The resulting XML document of the parsing process is stored for downloading. In future, it can be used for processing third party XML applications.
- E-Book
  The Open E-Book Initiative announced a standardized XML interface for E-Books [15]. Therefore, a utilization of the XML version of the study guide within an E-Book is possible. Because of the slight distribution of E-Books and in order to gain first experience for library applications it is planed to establish an E-Book loan procedure.

Print-basket

An innovative feature of the online study guide is the possibility of selecting single document parts. This functionality is realized by a so-called print-basket concept. Similar to the well known online warehouse-basket applications the user is offered a collection of documents or document parts instead of a sample catalogue. Within the web browser the desired document parts can be selected and gathered in the print-basket.

The implementation of the print-basket was realized in JavaScript. Therefore, in contrast to the widespread warehouse-baskets, based on cookie technology, the functionality of the print-basket is enabled independently of individual browser preferences. The actual definition of the print-basket is done by a JavaScript array:

```
var PrintBasket = new Array(TitleCount);
```

For every eligible document paragraph a binary value is written into this array.

The selectable parts of a document are combined with an HTML form button. By clicking the "Print" button the JavaScript method *onClick* is triggered and the corresponding parts are gathered in the print-basket.

```
<FORM>
...
<A NAME="pod.38"> </A>
<H1 CLASS="H1"> course of studies in print and media technology </H1>
<INPUT TYPE=BUTTON VALUE="Print" onClick="AddToPrintBasket(1)">
...
</FORM>
```

[2] Acrobat Reader is a trademark of Adobe Systems Incorporated

The HTML anchor element <A> and its attribute *NAME* is used as an unambiguous identification of the selectable parts of documents. For this, the attribute *select_id* from the study guide DTD is mapped on the HTML attribute *NAME*. The setting of the attribute value is done by the document editor while editing the XML documents. The unambiguous handling of the identification number among the XML and HTML document versions is controlled by a Perl script, which also inserts the "Print" buttons automatically.

After the selection process is finished the user is able to release the content of the print-basket to order his personal study guide. For the order mechanism another HTML form is used:

```
<FORM METHOD="post" ACTION="/Scripts/perl/pod_job.pl">
...
document.write('<INPUT TYPE="HIDDEN" NAME="pod_order"
VALUE="'+parent.PrintBasket+'">');
...
<INPUT TYPE=SUBMIT>
</FORM>
```

With the aid of the JavaScript method *document.write* the current content of the print-basket is transferred to the XML parser.

After a study guide order is received in the XML parser the individual document is assembled and structured according to the DTD and based on the current print-basket.

2.5. Editing Workflow

In order to edit a consisting study guide document or to create a new one and to provide it in the online study guide the following workflow is used:

1. Edit the XML documents
   As XML editor FrameMaker+SGML is used. Besides the XML document instances the DTD and the formatting rules can also be edited within FrameMaker+SGML.
2. Generate HTML documents
   To form an online HTML document from the XML document, first a temporary HTML version is produced by a simple HTML export out of FrameMaker+SGML. The resulting file, however, does not contain the above mentioned print-basket JavaScript constructs. The completation of the HTML document is done by a Perl application.
3. Update the study guide
   The modification of single documents in general causes modifications in other documents of the study guide. For example the table of content and the CGI scripts of the print-basket have to be updated. The corresponding update is also done by the Perl application.

The workflow steps 2 and 3 are run automatically. The editor administrates only the media neutral XML documents. Like this, redundant contents and the corresponding inconsistencies among the different media do not occur.

3. Generic Book Workflow

This section introduces a possible workflow for the production of “Generic Books“ as an extension of the existing study guide concept.

The Generic Book Workflow is designed to automate the publication process of a book from prepress to press and postpress. The idea is to produce booklets on the basis of digital distributed information resources, such as databases, organization and company websites, online-magazines, which leads to new ordered and structured information depending on the defaults of a user. This information will be automatically structured, layed out and finally printed as a bound book. The result is an individualized book according to the users demands.

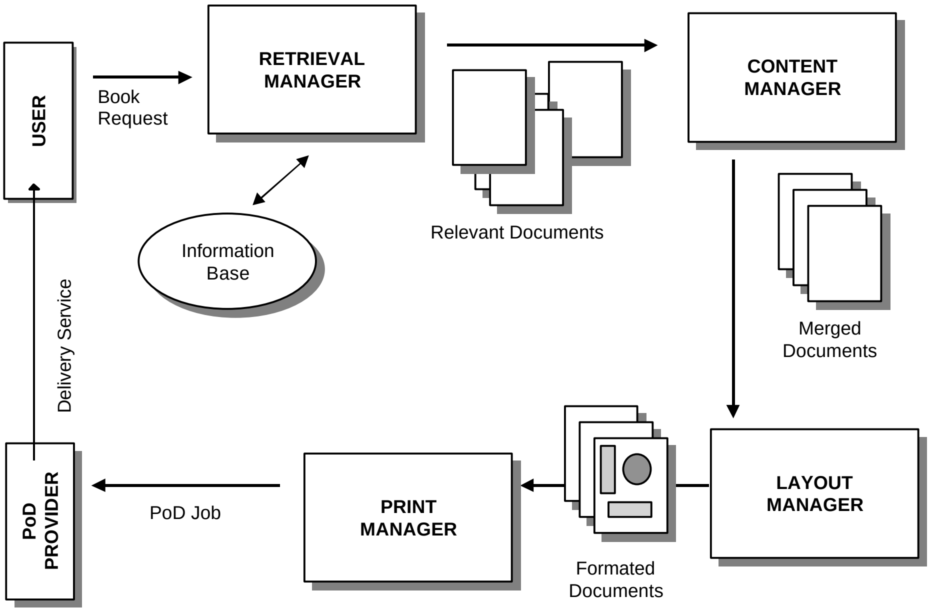

Figure 2: Generic Book Workflow

Figure 2 shows the overall workflow of a Generic Book production. At the beginning a user generates a book request with information about the desired content, the layout, the maximum costs, the book format, etc. Such a request will first be evaluated in the Retrieval-Manager, which transfers syntactically modulated inquiries to the proper search systems. The intention is to find as much content relevant sources as possible in the underlying information base. In the next step the different information sources will be arranged corresponding to their content. Depending on the generated order, typical book elements such as a table of content, an index, a table of pictures, etc. will be automatically created. The Layout-Manager is responsible for the machine-aided formatting of the whole document. The printable manuscript will be forwarded to the proper Print-on-Demand- (PoD-) Provider by the Print-Manager. On a high-tech print system with integrated in-line binding the single pages will be printed, gathered and bound. The finished book will be delivered to the user via an independent commercial service or via a PoD-Provider organised service.

The main extension of the study guide implemented currently is the integration of information sources with different structures. So far only document instances of a statically defined and therefore known DTD have been planned for the study guide. These requirements are not valid for the general Generic Book Workflow that will be implemented in an existing digital library. In general we face documents of different origin and therefore as well of different structure. An ad hoc use of such documents within one single book is not possible. First, there is no corresponding DTD neither for a flexible formatting of document fragments from different sources nor for a unified formatting of the generated overall document. For a future supplement of the existing study guide application into a complete Generic Book application two partial tasks are to be solved:

Firstly, a method is to be defined with which the structures of each document are identified automatically. Secondly, for the found several structures of the heterogeneous document collection a joint DTD has to be defined. In the last years some approaches in the field of automatic structure analysis have already been investigated: Y. Xu [16] elaborated a semi-automatic system which is learning structures of untagged documents from heuristic rules and user examples. In [17] and [18] methods to map common document fragments to well-known document elements such as sections, paragraphs, etc. are introduced. But the current state of the art does not allow a fully automatic structuring and assembling of unknown document collections. One of the problems is the ambiguity of automatic generated DTDs, i.e. resulting from incomplete documents [19].

Practical experiences concerning the merging of different DTDs into one overall DTD have been gained, among others, in the SuperJournal Project in the UK [20].

4. Conclusions

This paper introduced an online study guide that was implemented on the basis of XML documents. The approach offers an easy way to structure documents with respect to their content and provides a media neutral way to display and edit them. Important features of the study guide were presented:

- Individual Content
  The user is able to assemble an individual study guide out of the entire database and to affix personal annotations to it.

- Crossmedia Publication
  Based on the media neutral data management in a completely automatic workflow, an online as well as a print version of the study guide is produced. Furthermore, the study guide contents can be used for E-Books and third party XML applications.

- Online Print-basket
  A standard web browser serves as the user interface. Within the browser document fragments of the full version of the study guide can be arranged for an individual version.

The Generic Book Workflow was introduced as a continuing concept. This workflow allows the user of a digital library to get individual structured information in the form of a traditional paperbased book. Thereby, this approach preserves the advantages of the traditional library and merges them with the worldwide flexible access to the knowledge body of digital libraries.

The future developments of the study guide are orientated towards the Generic Book workflow. The aim is a realization of an automatic production of individual books on the basis of different document sources. The future emphasis in research and developing is:

- Use and development of existing methods to detect logical structures in unstructured documents automatically.

- Development of methods to merge different DTDs. The resulting DTDs should represent deep logical structures and offer a flexible sequential formatting of the elements included.

The development of new methods should be accompanied by implementations of a suited user interface that puts the new opportunities adequately at disposal. Among others, an extended print-basket is in preparation, in which the user can freely assemble his selected document fragments with intelligent software support - within the bounds of the DTD - and in which he can combine them with a personal layout.

References

1. J. R. C. Licklider. Libraries of the future. Cambridge, Mass (MIT Press) 1965.

2. Dillon, Andrew. Designing usable electronic text. In: Ergonomic aspects of human information usage. Taylor & Francis, London 1994.

3. Extensible Markup Language (XML). World Wide Web Consortium. Specification 1.0, 1998.(XML) 1.0. Available: http://www.W3.org/TR/ 1998/REC-xml-19980210 (6 September 1999).

4. Digital Library of the Goettingen Digitization Center of the Lower Saxony State and University Library. http://www.sub.uni-goettingen.de/gdz/en/gdz_main_en.html#dms (25 November 1999).

5. Interoperability and Intelligent Reuse of Distributed Teaching Materials. The TRIAL-Solution Project. The official link will be published by the European Commision: http://www.cordis.lu/ist/home.html (25 November 1999).

6. Project "Digital Dissertations" at the Humboldt University of Berlin, Germany. http://dochost.rz.hu-berlin.de/epdiss/index_en.html (25 November 1999).

7. Online Print-on-Demand Study Guide of the Institute for Print- and Media Technology. University of Chemnitz, Germany. http://www.pm.tu-chemnitz.de/sf/.

8. ISO 12083:1994, Information and Documentation - electronic manuscript preparation and markup. International Organization for Standardization, 1994.

9. DocBook 3.1. Organization for the Advancement of Structured Information Standards (OASIS). Released 1 Feb 1999. Available: http://www.oasis-open.org/docbook/docbook/ (6 September 1999).

10. TEI DTD. Text Encoding Initiative (TEI). Available: http://www.uic.edu/orgs/tei/ (6 September 1999).

11. MIL-STD-38784 (CALS DTD). U.S. Department of Defense Initiative: Continuous Acquisition and Life-Cycle Support (CALS). Available: http://navycals.dt.navy.mil/ dtdfosi/38784.html (6 September 1999).

12. HTML DTD. World Wide Web Consortium. HTML 4.0 Specification. Available: http://www.w3.org/TR/REC-html40 (6 September 1999).

13. D. Megginson. Structuring XML Documents. Prentice Hall PTR, New York 1998

14. C. Cooper. Using The Perl XML::Parser Module. Available: http://www.xml.com/xml/ pub/98/09/xml-perl.html (3 September 1999).

15. Open eBook Publication Structure 1.0 (Draft Version 14). Open eBook Initiative. Available: http://www.openebook.org/specfaq.htm (6 September 1999).

16. Y. Xu. An Incremental Approach to Document Structure Recognition. PhD Thesis. Technische Hochschule Darmstadt, Germany, June 1998.

17. 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 Compouter Science, University of Helsinki, 1996.

18. H. Ahonen. Generating Grammars for Structured Documents Using Grammatical Inference Methods. PhD Thesis. Report A-1996-4, Department of Compouter Science, University of Helsinki, Finland, 1996.

19. B. Klein, P. Fankhauser. Error tolerant Document Structure Analysis. International Journal on Digital Libraries, IJODL Vol. 1, No 4, 1997; Springer, pp. 344-357.

20. A. Apps, R. MacIntyre. The SuperJournal Project: Data Handling Using SGML. Proceedings of the 3rd Conference on Electronic Publishing EP'99, Ronneby, Sweden, May 1999, pp. 215-224.