# XML and the architecture of the Hyperbook

**Abdoulmajid Hakki**
Lappeenranta University of Technology
Department of Information technology
ahakki@lut.fi

*Abstract.* **This paper presents architecture for development of Hyperbook, using XML to describe the application domain to adapt the content of Hyperbook to the user's behaviour. XML data-centric orientation makes it possible to describe application domain, data access and dynamic data composition functions. The system architecture of the Hyperbook is summarised. The characteristic of XML useful to manipulate data in a dynamic way is described. A general approach to data representation is described and the metadata and data presentation is presented.**

## 1 INTRODUCTION

Since the emergence of the World Wide Web, the concept of hypertext has become a main representation and presentation format for variety of applications. Hypertext books or Hyperbooks are among them, which are characterized as a grouping of electronic text which considered as entity [1]. In most cases, these hyperbooks still retain the conventional book structure and are partitioned into sub documents called chapters, sections, subsections or appendices. In [2] have been identified four properties of constructive hypermedia. These properties include the following:

1. **Intertextuality**: the process of interpreting one text by means of a previously composed text.
2. **De-centeredness and re-centeredness**: the points of focus depend on the interactive learners forcing the active learning processes
3. **Multivocality**: the networking, multi-perspective, and multi-media features of hypertext. This includes the multi-perspective, multi-channel, and crisscross capabilities.
4. **Malleability:** allowing learners to transform the presentation of information into personal representations of knowledge

The linking mechanism of Hyperbook offers users freedom so that it becomes necessary to offer support during navigation. To efficiently allow the realisation of user-adaptable contents and presentations, a clear separation between multimedia contents, domain model and user model should be achieved [3].

Basic Component of Hyperbook Systems are:

- The Adaptation domain Model [3]
- User Adaptation Model
- Bookmark Model
- Storage Model [4].

This paper presents architecture for development of Hyperbook, using XML to describe the application domain and interface model to adapt the content of Hyperbook to the user's behaviour. XML data-centric orientation makes it possible to describe application domain, data access and dynamic data composition functions.

The rest of paper is organised as follows: Section two summarises the system architecture of the Hyperbook. Section three describes the characteristics of XML useful to manipulate data in a dynamic way. Section four describes a general approach to data representation. Section five and six present the approach to metadata and adaptive presentation.

## 2 SYSTEM ARCHITECTURE

The main goal of Hyperbook system is to provide customised view of the content of the book responding to different preferences, interest and users. The application is to serve the strategy of active readying as the need of society in information age. In this respect a system to support Application Domain and User Adaptation Model is designed. The system has a three-tier architecture comprising the *Author Module, Application* and *Data layer* (Fig. 1). *Author Module* corresponds to the browser. It allows designing and validating the XML documents to create the content of Hyperbook (2.1).

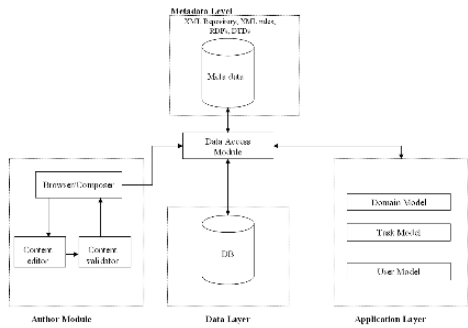


**Fig. 1:** Hyperbook system architecture - Application and Data Layers and Author Module.

The implementation of system is suggested either those concerning the on-line access to database, data composition and data delivery[1] or those allowing client-side elaboration (e.g. Java Applet).

### 2.1 AUTHORING MODULE

The Authoring Module (AM) comprise a content repository, which stores the content of Hyperbook files and an XML file which represent the structure of content.

The main object of Authoring Module is to allow design and validate (with respect to syntactic and semantic correctness the XML documents representing the concept of the Hyperbook. For simplifying the authoring process of Hyperbook the main components of Author Module are (fig. 1):

- The *Content editor (CE)*, which allows the content of Hyperbook.

[1]. E.g. JDBC, Java Servlet, Enterprise Java Beans, XML and SOAP .



*T. Sobh (ed.), Innovations and Advanced Techniques in Computer and Information Sciences and Engineering*, 63–66.

- The *Content validator*, which receives content description from CE and after validation of them, stores them in the Data layer (Data repository) [5].

### 2.2 APPLICATION MODULE

The task of Application layer is to provide integrated Hyperbook structure to the user. The application Layer consists of three main models:

- The *Domain Model* will be used to characterise the information accessible in the Hyperbook that is used to conclude in the user model.
- The *Task Model,* provide framework for structuring the content of Hyperbook and representing to user profile.
- The *User Model* stores individual users' information, preferences, goals and history.

### 2.3 DATA LAYER

The *Data Layer* (fig. 1) has two major means: the store of persistent data and to offer efficient access primitives. The content that stored to database consists of fine grained atomic units of the Hyperbook material. The aim of automatic units is to provide the maximum flexibility and reuse of these components in the variety of combinations [5].

### 2.4 METADATA LEVEL

The aim of using *Metadata* Level is to represent the logical structure of the data, as well as information concerning their usage and management (display, retrieval). A metadata is defined as a shared database of information about the content and data of the Hyperbook, that provided by the *Data Layer* or product by the Author. It stores: XML documents including the metadata, the Data Presentation Description, Schemes and XSL stylesheets [3, 5].

## 3 XML BINDING

Accessing Hyperbook from different devices will provide the basis for many important future in publishing industry. This will involve in some cases accessing local information that provide background about the local environment that are related to the content of Hyperbook. This service could be XML (extensible Markup Language) web service nature [6].

XML is a markup language for the construction of structured documents (fig. 2). It is a meta-language that allows us to create specialized markup languages for specific purposes [8]. XML is data-centric, which expresses the structure and eventually the meaning of the data contained in a document leaving its visualization to subsequent elaboration [9].

The Hyperbook application should be able to manage Modules from different types and domains of application using database technology which is suited to improve the usability and access to huge amounts of documents [10]. To make use of Hyperbook system, which architecture presented in previous sections, a format which allows storage, easy access, combination and adaptation of modules by a Hyperbook system is needed. The XML bindings are defined as a Document Type Declaration (DTD). The DTD definition was preferred to XML Schema [11].

The Hyperbook system should be able to manage *Adaptation Modules* from different types and domains of application using database technology which is particularly suited to improve especially the access to huge amounts of documents [12]. To make use of the advantages of our Hyperbook approach presented in the previous sections, we need a format which allows storage, easy access, combination and adaptation of *Modules* by a sophisticated system. XML document is a hierarchy comprising elements that have contents and attributes, XML is perfectly suited for representing the Hyperbook content, which is the conceptual content of the basic modules as well as the modular structure. The Hyperbook-specific model and its domain-specific instantiations serve as a well defined basis for a corresponding XML based markup language, the elements of which represent conceptual *Content Objects* by syntactical means. Furthermore the realization of View Specific by Style Sheets of the Extensible Style Language (XSL) [13] is used to Module Properties and Conceptual properties coming from XML bindings of different metadata proposals and standards as affirmed in [14].

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE ContentML SYSTEM "ContentML_simplified.dtd">
<ContentML>
      <ContentItem>
            <ContentItemId>uniqueId1234ABCD</ContentItemId>
            <Creator>tekija@domain.com</Creator>
            <DateTime>20060210</DateTime>
            <Language>fi</Language>
            <Category>Esimerkki sisällöt</Category>
            <Title>Asiasana merkkaus esimerkki</Title>
            <ContentData>
                  Tekstiä josta <Keyword>asiasana</Keyword> on merkattu.
                  <Keyword> Joukko erilliä asiasanoja
                        <KeywordAlias Value="asiasana"/>
                  </Keyword>
                  voidaan merkata siten että joukolle määritellään
                  yksi tai useampi alias.
            </ContentData>
      </ContentItem>
</ContentML>
```

**Fig. 2.** The XML content of the Hyperbook, created for mobile environment[2].

## 4 DATA REPRESENTATION

XML will play two distinct roles in Hyperbook development: The first role is that the infrastructure in which XML is used for data interchange and description. For example the content of book chapter are encoded in XML using SOAP and described in XML using WSDL (Web Services Description Language) [15]. The second role that XML plays could be named programming medium. In this programming role, the development of Hyperbook will be exposed to XML when designing the application interfaces in a top-down approach [15]. XML has several attractive features as a data representation and communication language, especially for Web applications.

The advantage of the XML is that it gives structure to unstructured data. It does this by embedding metadata tags in the data. Without these metadata tags, the fluid, unstructured contents of text documents would be difficult for Hyperbook application to

[2] The figure is part of Hyperbook project in Lappeenranta University of Technology for mobile environment, where six groups of students are participated. Figure 3. is one of examples, have been developed by group of students.