Adaptive Presentation of the Educational Module Information using XML

Mária BIELIKOVÁ

Slovak University of Technology
Department of Computer Science and Engineering
Ilkovičova 3, 812 19 Bratislava, Slovakia
bielik@elf.stuba.sk, http://www.dcs.elf.stuba.sk/~bielik/

Abstract. Educational module information indicates module characteristics and enables to answer such questions as "What is this module about?" or "Who is the reader of this module?" In this paper we describe the proposed framework for representation and adaptive presentation of the module hypermedia information. It is based on the XML (eXtensible Markup Language). Our work deals with adaptation of provided information during the module running. Adaptation is based primarily on the context of presentation. We experimented with the proposed approach and implemented a software prototype, which enables time view and user view of the module information. All presented data are stored in a database. Each element has associated constraints at the level of presentation, intended group of readers and presentation time represented by XML.

1 Introduction

The explosive growth of the World Wide Web as a medium for information dissemination caused several new problems, which have to be solved. The problems are often linked to continuously increasing amount of information that we are faced with. In this paper we deal with a problem of adaptive presentation of information represented as hypermedia.

Issues of adaptive presentation have received attention of the Web community only recently. Adaptive presentation is one of the generally recognised ways of adaptation in adaptive hypermedia [5]. It uses some data about the user (e.g., his knowledge about a particular problem, his previous visits) or about the context of the presentation (e.g., time) to present the content, which is the most interesting for the user. Another way of adaptation in hypermedia is adaptive navigation, which focuses on link adaptation.

In the process of design a method of adaptive presentation an application area should be considered. In our case we concentrate on educational hypermedia. The shift is away from a relatively static, localised paradigm of teaching and learning towards an interactive, dynamic, and non-localised paradigm [2]. Educational hypermedia typically deal with relatively small hyperspace. For adaptation they use the fact that students have often different goals and strategies of learning. A piece of educational material is often unclear for a novice and at the same time trivial and boring for an advanced student.

Methods and techniques of adaptive hypermedia are largely designed and evaluated within educational environments. They often focus on the educational materials, e.g. adaptive hypermedia books [6]. Summary of methods and techniques used in adaptive hypermedia (including educational) is given and discussed extensively by Peter Brusilovsky [5]. Described methods are based on a user model. They adapt the content and/or the navigation according to the user characteristics and requirements. Irrelevant information and links that overload the user's memory and screen is hidden, labelled as not recommended, or removed. Often additional information related to recommendations is presented by different colours [8].

Only little attention is devoted to delivering information about a module, its objectives, reader, lab requirements, schedule of lectures, conditions to obtain a grade etc. This kind of information is necessary for any kind of education delivery. Usually it is presented on the paper (on whiteboards at the departments), or electronically in the form of non-structured document files or as HTML hypertext document on the Web.

The content of the presented module information is usually solid. Each overlooked word can cause student later problems to fulfil requirements to obtain a grade. Although reading all available information can be useful in some cases, we have experience that for most students it is helpful to see details when they are current (i.e., needed). Such information can be represented by a presentation context. The presentation context is defined as a collection of data, which depends on the current state of the presentation but not on the data about the user.

Appropriate module information depends more on time of information reading (the context) instead of a detailed user model based on the knowledge related to the particular module content. Adaptation presented information according the current context would improve presentation and administration of such information.

The context model does not depend on the user model. For the purposes of module information, the context can be reduced to consider time of information reading. The context can contain also data about used browser or language. In this case we need several variants of the presented document and have to deal with the problem of hypermedia document version management. Methods and techniques for version management are discussed elsewhere [1]. Variants of content, which correspond to the particular user model, can be also considered.

2 Representation of the Module Information

The module information is represented by XML (eXtensible Markup Language). XML is a data format suitable for storing structured and semistructured text intended for dissemination and ultimate publication, perhaps on a variety media [4]. XML document contains chunks of the required information (elements) named and stored separately. Using XML allows interpretation of data of different documents without needing to adapt the application (i.e., interpreter of XML module data).

Each element has defined adaptation conditions in the form of required values of defined attributes. Elements can form an arbitrary hierarchy and can be named by the user (who is responsible for the module development). Presentation of information copies this hierarchy. In order to present 'pages' with reasonable length, some elements with long text and/or many offsprings can be displayed by the use of links.

Granularity of a document (the degree to which element's content is organised into offspring elements) is determined by a developer. The presentation method only requires to declare conditions (as predefined attributes of elements) and content of elements. Some elements however, do not have content in the form of text – they just name the hierarchy. This kind of elements is useful when the presentation is divided into several pages.

Each module has its own Document Type Definition (DTD) which defines the elements allowed in a particular document of module type. Several modules can define common DTD.

Bellow is an example of part of DTD declaration:

```
<!DOCTYPE Module [
  <!ELEMENT Module (Header, Body, Foot)>
  <!ELEMENT Header (Module-name, Module-Abbreviation?,
                    Study-Type?)>
  <!ELEMENT Module-name (#PCDATA)>
     …
  <!ELEMENT Body (Info?, Objectives, Syllabus, Labs, Faq?)>
     …
  <!ELEMENT Foot (Modification-date, Administrator-name?)>
…  ]>
```

A major decision in producing adaptive hypermedia systems is the decision about structuring information in such a way that it will be possible to adapt presentation. In our approach, the XML element according to the module data type definition is the smallest information chunk that can be adapted. However, the structure of the module information and the structure of the content to be adapted and presented can be of different granularity. Developer can define tags, which reflect the structure of presented information (e.g., paragraph tag `<p>`), but are not related to the structure of the module information.

Actual data related to the particular module are represented by the XML document according to its DTD. Each element contains several conditions, which define the model of presentation according to a given context. Attributes and their values within each element represent conditions. We have experimented with attributes related to the type of user, granularity of presentation, time and splitting of the presented information to several pages. Some of the used attributes are described in the next section. Set of available attribute-value pairs can be enhanced. In this case, their interpretation should also be defined.

3 Adapting the Module Information

We use the conditional text technique for adaptation [5]. With this technique, all information about a module is linked to the condition at the level of a user model or context model. When presenting information in the particular context and for the particular user, the system presents only information with the condition evaluated to be true. A simple example is hiding detailed information about the lab exercises in the first week of semester. The purpose of hiding this information is not to disable view of the information but prevent information overload in the first week of semester. Solution to this requirement is to disable some information in short or medium views in the first week, but enable it in the full view. Later, when information is current, it is presented also in short or medium views. This technique requires to set all the required conditions by administrator (or teacher) of the module. When the presentation of the module information is required, attributes of each element are evaluated and compared against the current context and adapted module information is displayed.

We don't use link removal for one concept. The main reason for this decision is the purpose of the system: presentation of administrative information related to the module. For this purpose, hiding or link/ content removal after "reading" something is not necessary.

We have developed a software prototype for adaptive presentation of a module information [7]. The software prototype uses the following data about the current state: group of the user, granularity of presentation and context represented by the current time. In order to reduce the amount of information displayed at once, two additional attributes are related with each element – *link* and *include*. Values of these attributes determine creation of links and splitting of some information into several parts presented independently and reached by hypertext links. Important simplification of representation of conditions concerning the current state is achieved by ordering possible attribute values representing the current state.

Bellow is an example of meta-data related to the element `syllabus`:

```
<syllabus description="Syllabus"
          from="01/01/2000" to=""
          user_operator=">=" user="guest"
          mode_operator="=" mode="medium"
          visible="yes" link="yes" include="no">
      <B>Background:</B>
      A brief history of expert systems; the expert system
      concept;
      ...
</syllabus>
```

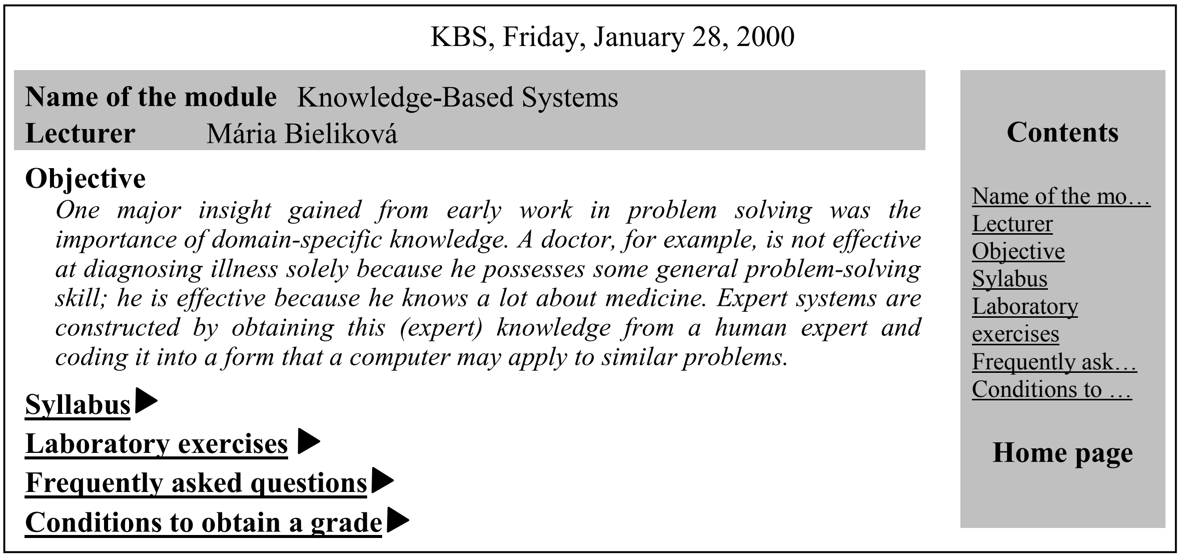
All meta-data defined in the element `syllabus` are evaluated according to the current state. If date of presentation is after the 1st of January 2000, the element `syllabus` is presented to any user (*user* >= *guest*) who have selected the medium granularity (*mode* = *medium*). When the element is presented, the text “Syllabus” is displayed. This text is marked as link (*link* = *yes*), which results in hiding the actual text of syllabus into the separate page (reachable through this link). When the link attribute is set to “no”, all text defined for this element is displayed.

Within our tool information about the modules can be presented in a consistent form. For example, all pages include a standard tool bar for navigating to the particular sections of the information. Form consistency is achieved by generating pages according to the defined style, which determines a page layout. Developers can combine text with the HTML tags in the content of the particular element.

The system recognises two types of links (similarly to [3]):

- links to the absolute URL (external) are represented by `<A>` tag with `HREF` attribute in the content of any element,
- conditional links are investigated by comparing the current state to actual representation of a particular element.

Figure 1 illustrates the part of appearance of presented information for the module Knowledge-Based Systems. Each page starts with a generated header, which contains text defined in the first element named `module` (the only parent of the hierarchy) and current date, and is automatically included in every page. The page content is generated according to the XML definition of the module information and the current context. Each page



KBS, Friday, January 28, 2000

Name of the module Knowledge-Based Systems
Lecturer Mária Bieliková

Objective

One major insight gained from early work in problem solving was the importance of domain-specific knowledge. A doctor, for example, is not effective at diagnosing illness solely because he possesses some general problem-solving skill; he is effective because he knows a lot about medicine. Expert systems are constructed by obtaining this (expert) knowledge from a human expert and coding it into a form that a computer may apply to similar problems.

Syllabus▶
Laboratory exercises ▶
Frequently asked questions▶
Conditions to obtain a grade▶

Contents

Name of the mo…
Lecturer
Objective
Sylabus
Laboratory exercises
Frequently ask…
Conditions to …

Home page

Figure 1. Example of module information presentation.

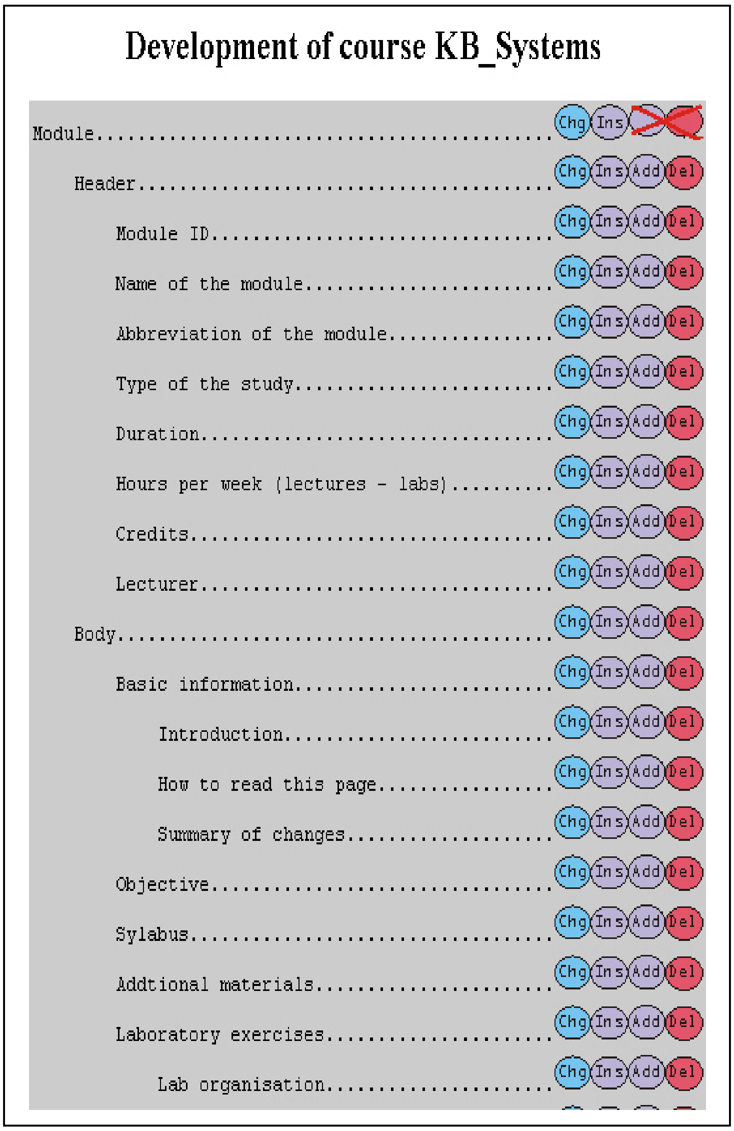
contains frame on the right hand side, which serves as a content navigation within the current page. Different data formatting can be expressed by a style sheet. Alternative style sheets can be applied at any time, changing the format to suite intended audience (e.g., format of the university web pages) and the capabilities of the used publishing media. Pages are assembled from these parts by means of PHP3 scripts.

We also developed a tool for supporting the design of adaptive hyperdocument, including definition of elements, conditions of their adaptation and the content. A developer can create and modify hierarchy of elements and define meta-data and the content of each element.

Figure 2 illustrates part of the hierarchy for the Knowledge-Based Systems module. On the left side the names of elements are displayed. A developer of the module uses the buttons on the right side, which serve for content modification (Chg), inserting a new element into the next hierarchy level (Ins), adding a new element into the same level of the hierarchy (Add), and deleting element from the hierarchy (Del).

Information related to a new module can be developed by the use of predefined templates or from the scratch. The element `module` is the starting element, which cannot be deleted. This element is the only element on the first level of the hierarchy (action add element on the same level is also not allowed). An experienced user can view and modify the XML code of module information.

Design and implementation of a prototype was oriented mainly toward prototyping the proposed framework for adaptive presentation of the module information. However, an effective use of such system in the Web environment requires effective storing and manipulation of the data about several modules, the data about users, and also requires some level of security (mainly at the level of authentication for modification of XML documents related to the particular module). Due to aims of the prototype development, the problem of security was not solved. Data manipulation efficiency is tackled by the design and implementation of relational database, which serves for storing the whole content of data elements together with the data about users and developers. We used the database server MySQL.



Development of course KB_Systems

```
Module
    Header
        Module ID
        Name of the module
        Abbreviation of the module
        Type of the study
        Duration
        Hours per week (lectures - labs)
        Credits
        Lecturer
    Body
        Basic information
            Introduction
            How to read this page
            Summary of changes
        Objective
        Sylabus
        Addtional materials
        Laboratory exercises
            Lab organisation
```

Figure 2. Example hierarchy of elements.

4 Conclusions

The paper is focused on the problem of adaptive presentation of organisational aspects of an educational module. It investigates three possible adaptive ways to help the users: hiding information according to the current time, considering type of a user and choice of the granularity of information (time, user and info views).

We use the conditional content adaptation technique for adaptation, similarly like the AHA system [3]. Unlike AHA system, where conditionals are encoded in the structured HTML comments, we represent conditions as attribute-value pairs in XML. Conditions are expressed declaratively as values of attributes. They are interpreted in our prototype by the PHP3 scripts. In [8], a similar approach is used, where constraints are expressed in the form of typed first-order logic, which allows for expressing complex conditions declaratively.

The main contribution of our work is the proposal of a flexible representation of an educational module organisational data using XML and considering the current context in adaptation of presentation.

An important feature of the good administrative module system is not only the adaptive presentation based on the current context, but also keeping modifications introduced to the pages. In traditional learning environments, the modification of the module material is not frequent and usually not interesting for a student (student receives package with learning material, and modifications are usually incorporated into a new version of learning material used in the next module run). A modification is frequent and important in presentation of organizational aspects of the module. The proposed representation of the module information allows support of modification management. We work on extension of the adaptive presentation, which will also consider recent modifications of the module information.

Acknowledgements

The work reported here was partially supported by Slovak Science Grant Agency, grant No. 1/7611/20.

References

[1] M. Bieliková and P. Návrat. Modelling Versioned Hypertext Documents. In: Proc. of System Configuration Management, SCM-8 Symposium, Brussels, Belgium, July 1998. B. Magnusson (Ed.), Springer Berlin, LNCS 1439, pages 188-197, 1998.

[2] C.M. Boroni, F.W. Goosey, M.T. Grinder and R.J. Ross. A Paradigm Shift! The Internet, the Web, Browsers, Java, and the Future of Computer Science Education. *SIGCSE Bulletin*, (1998) 145-152.

[3] P. De Bra and L. Calvi. AHA: A Generic Adaptive Hypermedia System. In: Proc. of the 2nd Workshop on Adaptived Hypertext and Hypermedia, Hypertext'98, Pittsburg USA, June 1998. Available on line from http://wwwis.win.tue.nl/ah98/DeBra.html.

[4] N. Bradley. The XML companion. Addison Wesley, Harlow, England. 1998. 440p.

[5] P. Brusilovsky. Methods and Techniques of Adaptive Hypermedia. *User Modeling and User Adapted Interaction* **6**, 2-3 (1996) 87-129.

[6] P. Brusilovsky and E. Schwarz. User as Student: Towards an Adaptive Interface for Advanced Web-Based Applications. In: Proc. of the 6th Int. Conf. User Modeling, A. Jameson, C. Paris, C. Tasso (Eds.), Springer Wien-New York, CSIM, pp. 177-188, 1997. Available on-line from http://um.org.

[7] L. Drobný. Web-Based Presentation of the Information about University Modules. Slovak University of Technology, Diploma Thesis, 1999. (in Slovak)

[8] P. Fröhlich and W. Nejdl. A Database-Oriented Approach to the Design of Educational Hyperbooks. In: Proc. of the workshop "Intelligent Educational Systems on the World Wide Web", 8th World Conference of the AIED Society, Kobe, Japan, 1997.

[9] G. Weber and M. Specht. User modeling and Adaptive Navigation Support in WWW-Based Tutoring Systems. In: Proc. of the 6th Int. Conf. User Modeling, A. Jameson, C. Paris, C. Tasso (Eds.), Springer, CSIM, pp. 289-300, 1997. Available on-line from http://um.org.