

WEBCOCT – A TOOL FOR PRODUCING WEB-BASED COURSES

Jolanta Berg, Anetta Brekiewicz, Joanna Jedrzejowicz, Jakub Neumann, Robert Suski

ABSTRACT

The aim of the paper is to describe the project WeBCoCT (Web Based Courses Creating Tools) intended to become a learning environment to help academic teachers to introduce web-based courses. WeBCoCT has as its main goal to support authors who do not have any experience with the preparation of web based materials. Some examples of courseware developed with WeBCoCT are provided.

KEYWORDS

Web based courses, XML technology, Learning Objects Metadata (LOM)

INTRODUCTION

The integration of computers in the learning process has led to the development of a large number of teaching and learning systems. They are often subject-oriented, developed and implemented to be used within one course at one university. Several examples of such systems were presented on previous CBLIS conferences. With the development of Internet technologies and e-learning courses offered online, the need for easily usable educational tools becomes even more pressing. Basing on our past experience (Jedrzejowicz, Kwapulinski and Kwapulinski, 2001) with developing subject oriented systems we started the WeBCoCT project, which is intended to become a flexible, extendable and easy to use learning environment allowing easy introduction of Web based courses by faculty members.

The idea of developing WeBCoCT has emerged in response to the fact that many existing tools developed at various universities or educational institutions do offer neither satisfactory flexibility nor generality and usability needed to become practicable learning environment. Systems like, for example (S. Margarita, 2002), (U. Lucke, D. Tvangarian, H. Vatterott, 2002) have been developed either as dedicated tools and are constrained to particular software/hardware environment, or use a particular natural language within the interface.

In WeBCoCT, in order to achieve high flexibility and to separate the contents of the course from its appearance the Extensible Markup Language (XML) is used for document processing. The description of the document is splitted into its structure (XML Schema), the contents (described in XML) and an output transformation for presentation (described with the Extensible Stylesheet Language, XSL). The use of XML allows the user to be independent from hardware and software environment since any web server and browser may handle WeBCoCT. When designing the system much effort was made to achieve user-friendliness, thus allowing users not acquainted with the XML technology to set up their own courses fairly easily.

The structure of the paper is as follows. After presenting an overview of the system and its development we describe the architecture, discuss the benefits of using the extensible markup language for the project, and show examples of the application of our tool for building web based courses. The

concluding section includes details of the future improvements of the system as well as suggestions for further research.

OVERVIEW OF WeBCoCT – SYSTEM DEVELOPMENT

Developing an on-line system of tools from scratch was not attempted without careful preparation and planning. The first stage involved the requirement analysis resulting in the identification of users needs. This was used to define the scope, form, processes and users. Next step was to design the architecture of the system, as shown in Figure 2. Finally, an interface was developed that was simple and required minimal interpretation. It also needed to be extensible.

An existing Linux/Apache server was chosen to host the system. It rapidly became clear that the original server was unable to take the large quantity of traffic generated by the new site and a dedicated server was therefore acquired to specifically host WeBCoCT, it can be found under the address <http://WeBCoCT.math.univ.gda.pl>.

To describe the system we consider the following features: the users with their roles, and the data flow. Two groups of users can be distinguished, that is authors and students. Since e-learning applications are nowadays mostly Internet based for both teaching and content management functions, it has been decided that both groups of users will work with browsers to communicate with the system.

The authors use the template of the course (see Figure 1), offered by WeBCoCT. The template was designed in the first stage of the project. Each course is described by a set of metadata, which contains the following items: the name of the subject supported by the course, the author, keywords, and the date of the last modification. The form of the syntax and semantics of the above metadata is similar to Learning Object Metadata (IEEE Learning Technology Standards Committee, Draft Standard for Learning Object Metadata), which is considered a standard in the field of educational material. The contents of the course include tutorials, tests, interactive exercises, the questionnaire for the users, and past exams. In Figure 1 the dotted line marks the optional parts of each course.

The authors using our tools have the possibility of designing their own courses in terms of defining the type of tests, their number and contents; the same applies to tutorials, questionnaires and past exams. In case of exercises the situation is different. First of all, exercises are subject-specific. What is more, interactivity is the key concept of the project and the structure of exercises should reflect its importance. These are the reasons why the form of exercises is already suggested to the authors and they have to adjust to the given form. In EXAMPLE APPLICATIONS some example exercises are described.

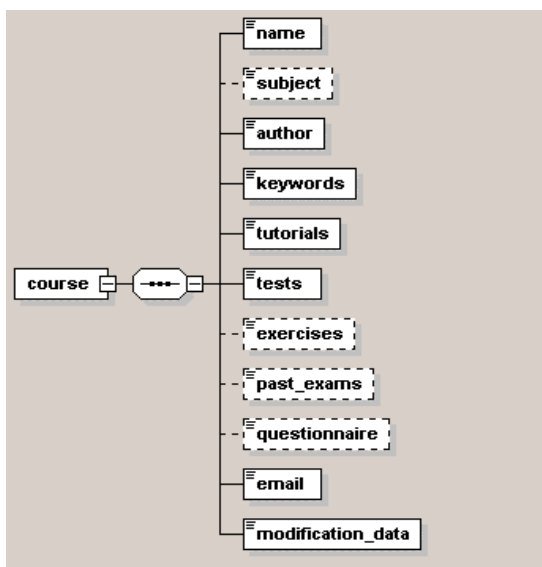


Figure 1. The template of a course

ARCHITECTURE OF THE SYSTEM

The system has a three-tier architecture (Figure 3) where the course design functions are separated from the authoring tools and teaching functions. The two groups of users of WeBCoCT that is authors and students have access to the presentation tier. The database tier contains XML Schema files describing the structure of the course and its parts suited for several forms of learning (tests, tutorials, interactive exercises, past exams, etc.), metadata containing information about the author, domain of the course, coverage, educational objectives and XSL files responsible for the presentation of each part of the course. The XML processing tier contains tools for documents parsing and transforming. It is possible to exchange components in this tier and use other tools, if desirable.

WeBCoCT runs under Unix operating system, makes use of web server Apache and the system Postgres for the database part. In the implementation process mark-up languages HTML and XML as well as programming tools and languages PHP, Java, Java Script, XML processors were used.

The database contains the information on users and courses for administration purposes that is restricting access to authoring tools only to a group of authors, and parts of courses only for students with logins and passwords – some parts of courses can be reached without any limitations.

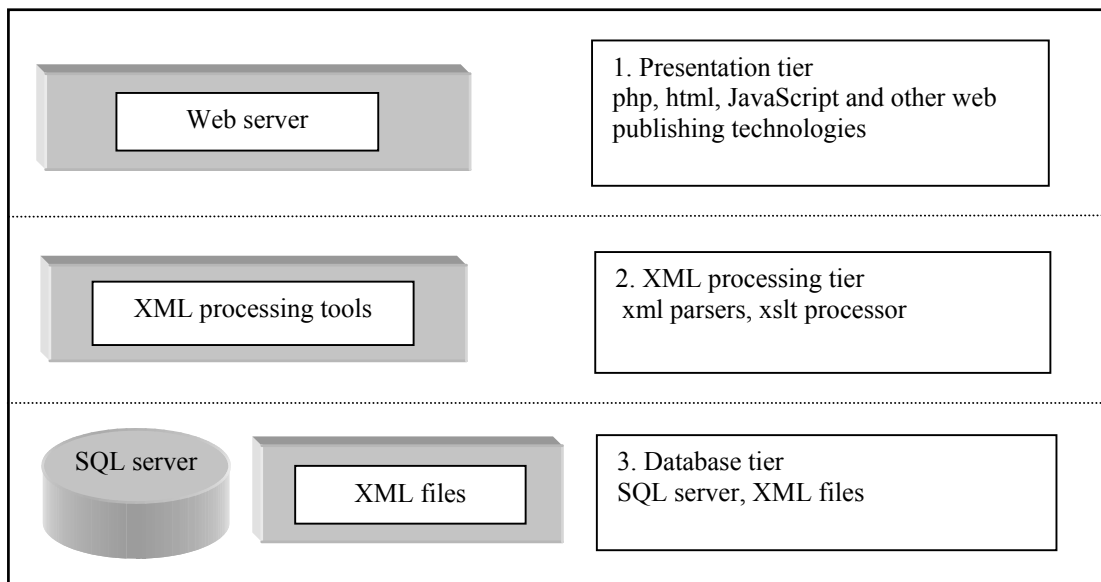


Figure 2. Three-tier architecture

WeBCoCT, as was already mentioned, XML Schema files are used to describe the logical structure of courses and their parts: tests, exercises, and tutorials. This allows for the distinction between the content of a document (in XML file) and its layout and allows to strictly forcing the structure of documents. Furthermore as the documents are marked-up with tags, tools (for example XML Spy) can automatically parse the documents to check whether they are well formed and valid (strict conformance to the structure constrained by XML Schema).

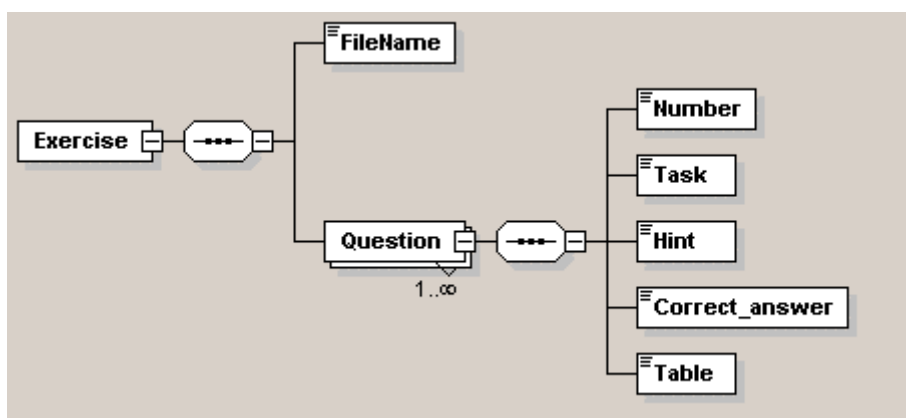
The task of introducing new versions of an existing course can now be reduced to editing appropriate XML files without any need for program change. For example, replacing an XML file containing a test by another one, will result in offering students new opportunities for self-testing. What is more, the author can perform this task easily from his computer if only it is Internet connected.

Probably the main advantage of using XML is the possibility to reuse data in different teaching situations – using the abstract XML description of a course new authors are able to compose individual courses from this material and introduce new features. For example, exam questions can be generated from tests and exercises. The same applies to preparing teaching materials (handouts, slides, presentations, etc.) used in face-face education alongside online teaching.

EXAMPLES OF APPLICATIONS

Two courses in computer science were prepared with the help of WeBCoCT – the course on databases that is a further development of (J.Jedrzejowicz, M. Kwapulinski, P. Kwapulinski, 2001) and a course on automata and formal languages. Below we concentrate on interactive exercises as they are core parts and are subject-specific, differing for each course.

In case of database course the use of user-friendly tools allowed to extend the group of authors since upgrading exercises is simple and easy. The author can either edit/extend already existing exercises, or introduce new ones. In both cases he/she works in a graphic environment and fills the forms, which allow the system to define an XML document. For example exercises teaching SQL require task formulation, SQL command (usually it is not unique), which performs the task, and probably a hint, which is optional. In Figure 4 we show an XML Schema document for this type of exercise and an example of a suitable XML file.



```

<?xml version="1.0" encoding="ISO8859-2"?>
<?xml-stylesheet type="text/xsl" href="Exercises.xsl"?>
<Exercise xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance>
  <FileName>exer1</FileName>
  <Question>
    <Number>1
  </Number>

```

```

<Task>Create the table Employee that contains the
following attributes:ID, name, surname, dept, insur_no.The
atrribute insur_no is boolean, all the other are strings.
</Task>
<Hint>Use CREATE TABLE command
</Hint>
<Correct_answer>Create table Employeeadmin(ID int primary
key,name varchar(20),surname varchar(20),dept varchar(10),
insur_no boolean)
</Correct_answer>
<Table>1
</Table>
</Question>
</Exercise>

```

Figure 4. The structure of an exercise

During an authoring session the author has an online access to all the views and tables already in the database, as well as dictionary tables. This helps to set-up the exercise, introduce new tables if necessary, and check the answer before submitting it into the system. An example of the author's session, when introducing the exercise on SQL, is given in Table 5. The author enters the following data into the system: the description of the exercise (metadata), the task for the student, a hint (if necessary), the correct answer, names of tables in the database, if applicable. The session is concluded with the generation of an appropriate XML file, as shown in Figure 4.

During the teaching session each student works on his own database. This allows students to split up their work and not necessarily do the whole job during one session. Since all the correct commands (introduced by the author) are performed on the mirror image of student's database, the results can be compared. Thus the user gets feedback from the system explaining the mistakes, if any.

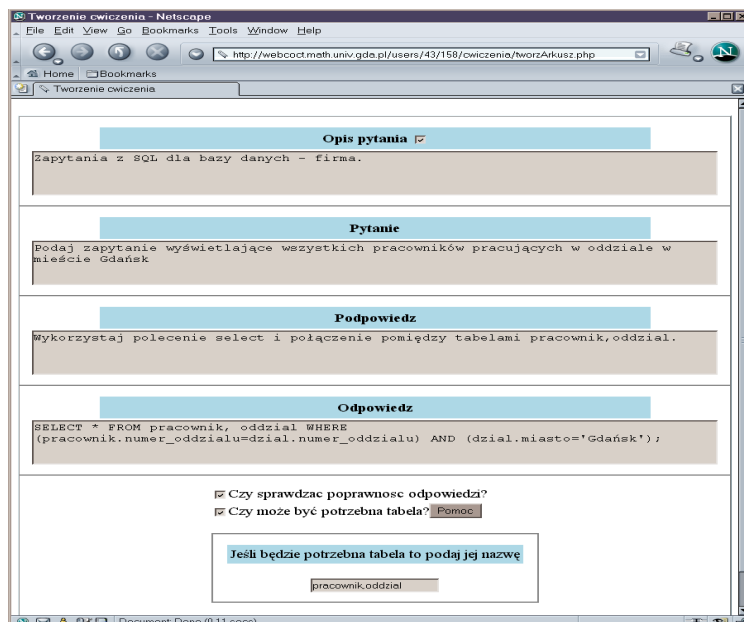


Figure 5. Author's session

The second course contains topics on finite automata and regular languages. The idea behind exercises is similar – to make most of the interactivity. The tasks for users are of type: *define an automaton such that..., describe the language of the automaton given by the graph, convert the given automaton into an equivalent one which ...,etc.* Though the answers need not be unique, it is possible to check their

correctness applying well-known procedures to the correct answers introduced during the authoring session.

CONCLUSIONS AND FUTURE WORK

The paper reports on the work-in-progress. The described software is still being tested. Its application to the development of courseware in computer science has shown it is a time saving tool for building web-based courses. Applications for language teaching are far advanced and we hope to give an account during the conference.

Further plans include:

- developing tools for easy publication of mathematical texts, for example allow for documents in widely appreciated TeX language,
- setting up a subject-independent template for exercises so that introducing exercises for a new course will only require developing a subject-specific module to check students answers,
- performing a survey among users-authors on how user friendly the system really is and how can it be still improved,
- manage feedback from users-students and using the students profile make suggestions concerning further work for single students.

REFERENCES

J.Jedrzejowicz, M. Kwapulinski, P. Kwapulinski (2001) – Postcourse -WWW-based course on databases, Proc. CBLIS'01 (paper E4),

U. Lucke, D. Tvangarian, H. Vatterott (2002) – The use of XML for the development of multimedia teaching and learning system, Networked Learning Conference, Berlin NL'02

S. Wiest, A. Zell (2001) – XML in web based training: workflow and opportunities, CBLIS'01 (paper E2),

S. Margarita (2002) – WEBGen : a batch generator of Web-based courseware, Networked Learning Conference, Berlin NL'02

World Wide Web Consortium (W3C) <http://www.w3.org/XML>

IEEE Learning Technology Standards Committee, Draft Standard for Learning Object Metadata, <http://ltsc.ieee.org/wg12/>

Jolanta Berg, Anetta Brekiewicz, Joanna Jedrzejowicz, Jakub Neumann, Robert Suski
Institute of Mathematics,
University of Gdansk,
ul. Wita Stwosza 57,
80-952 GDANSK,
POLAND,
Email: webcoct@math.univ.gda.pl