An XML-based Java Application for the Management of Online Questionnaires

Sabin-Corneliu Buraga, Mihaela Brut, and Daniel Onacă

***Abstract* — The paper details a flexible and extensible Web application focused on creation and management of online questionnaires. The solution is based on Java and the storage system adopts native XML databases. The application is a part of I*Teach project, in order to give teachers possibility to use a tool for organize a suite of tests, accordingly with their specialization.**

***Keywords* — E-learning, online testing, modular implementation, Java platform, native XML database.**

I. Introduction

THE e-learning is defined by "the use of new multimedia technologies and the Internet to improve the quality of learning by facilitating access to resources and services as well as remote exchanges and collaboration" [6]. Also, referred to as Web-based training, online learning, distributed learning, or technology for learning, the e-learning became a recognizable presence in the all kinds of education, including the public education endorsed by various companies through their guides and other types of electronic publications, promotional materials, FAQ (*Frequently Asked Questions*), collaborative portals, etc.

An important feature that an e-learning system must provide is a proper evaluation module in order to give students the possibility to self-personal development in the studied curricular area. Presently, there could be noticed a great importance being accorded to the open-source Web technologies regarding the online evaluation methods in a flexible environment. The current development in the e-learning domain conducted to a large amount of tools for easy management of online tests [2].

We shall present below a system based on XML (Extensible Markup Language) [1] and Java technologies for online testing, part of the I*Teach Leonardo da Vinci Project developed within Faculty of Computer Science, "A.I.Cuza" University of Iasi, Romania.

The paper is organized as follows. Section II gives some details regarding I*Teach Project and section III concerns the overall architecture of the developed Web system. The testing modules are implemented in Java and using native XML databases for the data storage. Last sections are dedicated to further directions of research and final conclusions of the paper.

This research work is partially supported by Innovative Teacher (I*Teach) Leonardo da Vinci project, BG/05/B/P/PP-166038.

S. Buraga is with the Faculty of Computer Science, "A.I.Cuza" University of Iasi, Romania (phone: 40/232/201090; fax: 40/232/201490; e-mail: busaco@infoiasi.ro).

M. Brut is with the Faculty of Computer Science, "A.I.Cuza" University of Iasi, Romania (phone: 40/232/201090; fax: 40/232/201490; e-mail: mihaela@infoiasi.ro).

D. Onaca graduated in 2005 the Faculty of Computer Science, "A.I.Cuza" University of Iasi, Romania.

II. General Presentation of the I*Teach Project

In the actual knowledge society, there are considered to be important the so-called "enhanced skills", such as information and knowledge articulation and presentation, information search, gathering, evaluation, and effective use, project working, team working, problem solving, and skills for life-long learning.

The aim of the *Innovative Teacher* project (*I*Teach*) [8] is to develop *practical methodologies*, *approaches*, and *tools* targeted at day-to-day utilization by the teacher trainers and teachers of these enhanced ICT skills in their work.

The target groups of the I*Teach project are:

- Tutors trainers in ICT from universities and teacher training institutions (pre-service and in-service).
- Tutors and trainers (both pre-service and in-service) in ICT-related subjects from vocational schools, specialized secondary schools, training organizations and departments.

The important concrete aims of the project are:

1. To develop a teacher trainers' and teachers' methodological handbook of practical methods, methodological tools, and software instruments to support students in building *enhanced ICT skills* and competences.
2. To design a teacher training curriculum for using the handbook methodology & teaching *enhanced ICT skills*.
3. To develop an online multilingual content repository with methodological and learning resources for teachers.
4. To create virtual training centers in project partners countries for continuous teachers' support and training.
5. To disseminate the project findings, results, and materials both nationally and throughout Europe.

The expected main project results are:

1. Developed methodological handbook, tools and instruments for *enhanced ICT skills* teaching.
2. Developed teacher training sample curriculum for applying the developed methodology.

3. Trained teacher educators and teachers to apply the developed methodologies.
4. An online multilingual content repository open to the European educational communities of practitioners, containing a variety of the handbook-model developed learning materials.
5. Established virtual training centers in participating countries, working sustainable beyond the project life-time, providing training and consultancy in ICT-teaching and maintenance and upgrade of the online repository.

The project consortium - which members are localized in Bulgaria, Germany, Italy, Lithuania, Poland, Romania, and The Netherlands - is composed to ensure sufficient and complementary expertise with respect to producing the planned project outcomes, and enough potential for European wide valorization of the project results. More details are available in [8].

The main responsibility of the Romanian team is to provide an efficient and flexible solution for management of the multilingual Web repository used for storage and manipulation of didactic materials in an intelligent way. Among the implemented tools, a testing Web application is already developed. This application will be detailed below.

III. Overall Architecture of the Testing System

A. System Requirements

In order to give users the possibility to use the system in a proper manner, the following requirements were addressed:

- all information must be stored in a (semi)structured way, by using XML technologies, such as native XML databases;
- the management of the native XML database must be available on the Web, for the authenticated users (tutors, students, and administrators of the system);
- providing a high-level of information security;
- the structure of the designed databases must be independent from involved processing tasks;
- an integrated framework for editing and managing of various questionnaires, regarding the database structure of the system);
- possibility to internationalize the application and to store multi-lingual resources;
- portability of the whole application, by exploiting it on various operating environments;
- using a modular approach in order to give possibility for further extensions of the implemented systems (for example, to support intelligent testing and evaluation of the students).

B. Modular Architecture

We choose to use a modular architecture in order to gain flexibility and to obtain extensibility. To assure platform independence, a Java-based solution of implementation is adopted.

The approach is using the classical client/server paradigm, in which the server offers all facilities for storing and managing defined questionnaires, and the clients - through a Web-based flexible interface - accessed the implemented services.

The following packages are developed (see figure 1):

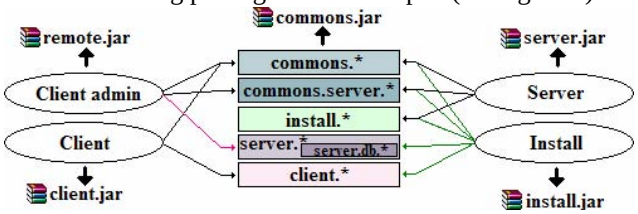

Figure 1. General structure of the testing system.

- *commons.** provides general data structures used by the rest of the packages and gives support for internationalization; this package includes the support for secure connections, by using SSL (Secure Socket Layer) version 3 and TLS (Transport Layer Security) version 1, implemented as specialized Java classes;
- *commons.server.** defines the classes which will be commonly used by the server, as well as by the administration client at the level of the server; a command-line client is provided for quick administration of the server, similar with a console tool;
- *server.** provides all functionality of the server (e.g., access to native XML databases, logging, user auditing and preferences); the structure of this package is depicted in figure 2.

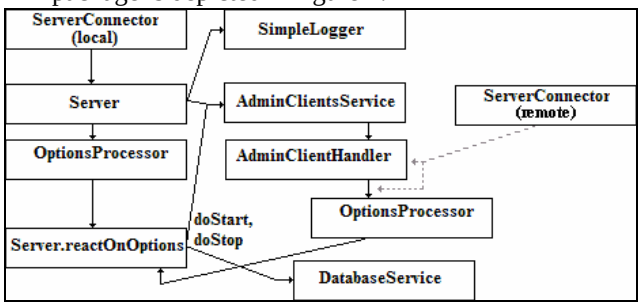

Figure 2. Classes implemented by *server.** package.

- *server.db.** is a sub-package of the previous one and implements the functionality of all services which deal with databases; the package gives support for the tutors clients. The package is responsible with validation of XML data provided by a client and with insertion/deletion of questionnaires into/from a native XML database;
- *client.** represents a generic client to be used by tutors in order to create, modify, and administer the test suite.
- *install.** is a package used to install all components of the system into a certain operating environment.

The sources of the developed Web application are encapsulated into several *.jar* archives (for example, *remote.jar* is dealing with all remote connections to the system, from certain Web clients) - see figure 1.

C. XML Database Structure

All information within system is stored as XML documents into a native XML database. An important

concept is the questionnaire. This XML document can have two forms:

- A general form - to be used to populate the questionnaire with questions imported from external data sources (such as, HTML pages, Excel sheets, etc.).
- A specific form - based on previous one, is used to map (and extend) a questionnaire for the purpose of being properly stored into a particular native XML database system.

The main components of a questionnaire are:

- *Question body* signifies the question content (mandatory); it can be used any valid XML construction (such as, XHTML elements, SVG shapes, SMIL presentations, MathML equations, etc.);
- *List of possible responses* - for each response a number of assigned points can be defined (e.g., positive numbers for each correct response);
- *Motivation of choosing the right response* (if any); the content of this element could also be any valid XML construct;
- *Question difficulty* (such as, easy, medium, hard).
- *Test category* defines the topic of the given test (e.g., Web programming, a subtopic of Web technologies).

Another type of XML document concerns the tutor (personal data, preferences, taught disciplines, etc.). Of course, to model the student profile, a specific XML document is used.

All XML documents are validated via corresponding XML schemas, according to XML Schema [4] specification.

D. Implementation

The Web application is using the following Java technologies:

- *Java 2 Enterprise Edition 1.5* (J2EE) provides the programmatic aspects that concerns this project;
- *Java Secure Sockets Extension* (JSSE) is a Java API used for implementation of SSL and TLS connections (gives support for user authentication, data security, message integrity, etc.);
- *Java Native Interface* (JNI) assuring the inter-operability with other APIs written in other programming languages (is used for connection to database XML native server);

For storage, the Berkeley DB XML [5] system is used. The server is available as open source software and provides facilities for processing and querying XML documents and collections via DOM, XPath, and XQuery - details in [10].

All tests and deployment tools are available on Apache Web server.

An example of administrative login via a Java graphical client is provided by figure 3. The management of the questionnaires categories is depicted in figure 4.

The system can be easily configured in respect with internationalization. We give a fragment from the file storing the system messages in English language (similar, could be chosen other languages):

```
server.msg.asknewpwd = Input new server password:' '
server.msg.askoldpwd = Input server authentication password:' '
server.msg.confirmnewpwd = Confirm new password:' '

server.err.send = error sending data to server ({0})
server.err.notd = nothing to do
server.err.export = can"t export settings{0}
server.err.import = can"t import settings{0}
server.err.receive = invalid data received from server ({0})
```

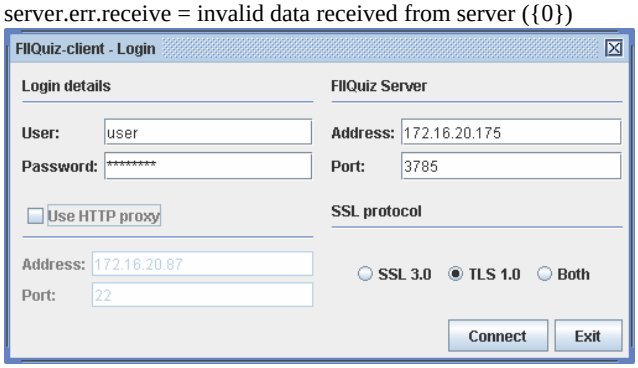

Figure 3. Web interface of the login task.

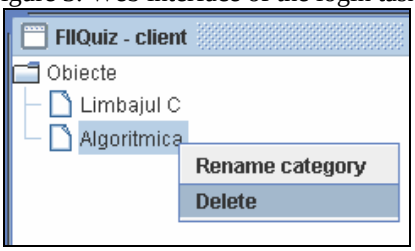

Figure 4. Browsing questionnaires categories by internal querying the native XML databases server.

The insertion of a question into a certain questionnaire is easily facilitated to a tutor by a Web interface showed in figure 5 (this Web interface is provided in Romanian language).

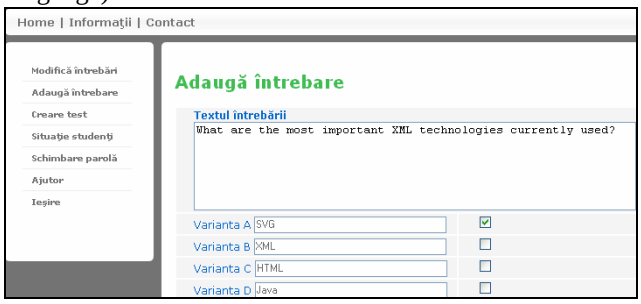

Figure 5. Adding a question to a questionnaire

IV. Related Approaches

There are many other systems to be used in the process of student evaluation. We present two important types of applications.

A high-cost suite is *Perception* [9], produced by *Questionmark* Company. This application permits creation of different questionnaires by using diverse user-interfaces (Windows application, browser or Microsoft Word). Evaluation process can be easily administered (the program provides support for schedulers, user groups and courses). The questionnaires could be filled in via different ways (by using Web browsers or integrated network applications). This product does not using a database for questionnaire storage, all information being stored as regular files. Questionnaire editing tools are not

portable. Some security requirements are missing.

A low-cost tool is *Hot Potatoes* [7], provided by *Half Baked Software Inc.*, which enables the editing and managing of different types of questionnaires to be published on Web. The system can be freely used, in a limited manner, within the academic environments. The questionnaires are stored in a proprietary format, but could be exported as HTML+JavaScript files. The users are not authenticated and the results of tests are not stored on a server. The tool is suitable only for self-evaluation.

V. Conclusion

In this paper we presented a flexible and extensible Web application focused on creation and management of online questionnaires. All used technologies are freely available and the storage solution adopts native XML databases.

The application can be considered as a part of I*Teach project, in order to give teachers possibility to use a tool for organize a suite of tests, accordingly with their specialization.

Modular structure and versatility provided by XML technologies (see section III) make a suitable candidate for an effective deployment in an academic environment.

Further directions of research will focused on conceptual modeling of questionnaires knowledge domain and the semantic relation between questionnaires, didactic materials, tutors, and students. This approach can present an interesting exploration in the context of semantic Web technologies.

Another approach is to integrate the presented application into a multi-agent system, following the guidelines presented in [3].

Appendix

An XML Schema fragment is provided below for validation of the general form of a questionnaire:

```
<xsd:schema
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    targetNamespace="http://thor.info.uaic.ro"
    xmlns:FIIQuiz="http://thor.info.uaic.ro"
    elementFormDefault="unqualified">
  <xsd:include schemaLocation="externalQuestionType.xsd" />
  <xsd:complexType name="questionType">
    <xsd:complexContent>
      <xsd:extension base="FIIQuiz:externalQuestionType">
        <xsd:sequence>
          <!-- number of effective utilization -->
          <xsd:element name="usage" type="FIIQuiz:usageType" />
          <xsd:element minOccurs="0"
                  name="excludedQuestions"
                  type="FIIQuiz:excludedQuestionsType" />
        </xsd:sequence>
        <xsd:attribute name="ownerID" type="xsd:token"
                  use="required" />
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>

  <xsd:complexType name="usageType">
    <xsd:attribute use="required"
              name="activeCount"
              type="xsd:unsignedInt" />
    <!-- number of tests that use this question -->
    <xsd:attribute use="required"
              name="inactiveCount"
              type="xsd:unsignedInt" />
  </xsd:complexType>

  <xsd:complexType name="excludedQuestionsType">
    <xsd:sequence>
      <xsd:element maxOccurs="unbounded"
              name="excludedQuestion"
              type="FIIQuiz:excludedQuestionType" />
    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="excludedQuestionType">
    <xsd:attribute name="ID" use="required"
              type="xsd:token" />
  </xsd:complexType>

</xsd:schema>
```

References

[1] T. Bray et al. (eds.), *Extensible Markup Language 1.0 (Third Edition)*, W3C Recommendation, Boston, 2004: http://www.w3.org/TR/REC-xml

[2] M. Brut, *Instruments for E-learning* (in Romanian), Polirom, Iasi, 2006 (to appear)

[3] S. Buraga, "Developing Agent-Oriented E-Learning Systems", *Proceedings of The 14th International Conference on Control Systems And Computer Science – vol.II*, I.Dumitrache and C.Buiu (eds.), Politehnica Press, Bucharest, 2003

[4] D. Fallside, P. Walmsley (eds.), *XML Schema Part 0: Primer Second Edition*, W3C Recommendation, Boston, 2004: http://www.w3.org/TR/xmlschema-0/

[5] * * *, *Berkeley DB XML*: http://www.sleepycat.com/

[6] * * *, *eLearning Europa.info - An initiative of the European Commission*, Glossary: http://www.elearningeuropa.info /index.php?page=glossary

[7] * * *, *HotPotaoes*: http://web.uvic.ca/hrd/hotpot

[8] ***, *I*Teach Project*: http://i-teach.fmi.uni-sofia.bg

[9] * * *, *Question Mark's Perception*: http://www.questionmark.com/us/perception

[10] * * *, *World-Wide Web Consortium's Technical Reports*: http://www.w3.org/TR/