DOCUMENT RESUME

ED 444 585		IR 020 241

AUTHOR	Ahern, Terence C.; Jamison, Mark; Olivarez, Arturo
TITLE	Designing the Next Generation of Tools Using an Open Systems Approach: A Usability Study.
PUB DATE	2000-00-00
NOTE	7p.; In: Society for Information Technology & Teacher Education International Conference: Proceedings of SITE 2000 (11th, San Diego, California, February 8-12, 2000). Volumes 1-3; see IR 020 112. Figures contain illegible type.
PUB TYPE	Reports - Research (143) -- Speeches/Meeting Papers (150)
EDRS PRICE	MF01/PC01 Plus Postage.
DESCRIPTORS	*Computer Assisted Testing; Educational Technology; Higher Education; *Instructional Design; Instructional Effectiveness; Preservice Teacher Education; Student Attitudes; Student Surveys; Systems Approach; *Test Construction; Test Format; *World Wide Web
IDENTIFIERS	HTML; *Open Systems Theory; Technology Integration; *Usability; XML

ABSTRACT

This paper reports on the results of a usability study that investigated the suitability of using an open systems approach for developing an online assessment application. Issues examined included ease of use, security, effectiveness, and ease of maintenance. Participants were 68 preservice teachers taking a technology integration course. Students were administered a Web-based demographics survey, followed by an exam of 25 multiple choice questions. Results indicate that an open systems approach was a feasible way to design instructional tools. Data encapsulation using XML (Extensible Markup Language) was easily created, maintained, and modified, allowing for a seamless interaction on the part of the faculty. The students reported in exit interviews that taking the exam was easy, and they liked the convenience of immediate scores. (Contains 11 references.) (MES)

Designing the next generation of tools using an open systems approach: A usability study

Terence C. Ahern
Mark Jamison
Arturo Olivarez
Texas Tech University
dwtca@ttacs.ttu.edu

Abstract. Educational technology is at a crossroads. The Internet was originally designed for the needs of the science and military, not for the purposes of sound instruction. New and more powerful protocol schemes are needed to truly realize the potential of networked-based instruction. A new technology that holds a lot of promise is XML. XML provides a more structured and dynamic approach to content creation that allows the developer to define and order data instead of merely displaying it. We were interested in some of the issues in using a new technology such as XML for instructional applications. In this study we found that the Internet and XML were suitable for this type of application. We found no difference in performance from either local accessed classes and those delivered from an Internet server.

Introduction

Educational technology is at a crossroads. Rapid development of computer technology has provided the individual with an unprecedented access to information and resources that only a generation ago was the stuff of science fiction. Never in our history has technology been so important in the success or failure of our students. However, the Internet was originally designed for the needs of the science and military, not for the purposes of sound instruction. Consequently, newer tools and protocols are needed that truly facilitate teaching and evaluation and put into the hands of teachers the power to design. One area that is critical to student success is student assessment. This paper reports on the results of a usability study that investigated the suitability of using an open systems approach for developing an online assessment application. We were interested in the following issues: ease of use, security, effectiveness and ease of maintenance.

XML and data encapsulation

The pioneers of the Internet needed a blueprint for a system that relied on a simple message format in conjunction with simple interactive software. They created a system that would work even though they had incompatible computer resources. An environment evolved that fostered interoperability and sharing at the hardware, software and human levels" (Berger, 1999 p. 27). Secondly, they created standard interaction protocols that were "independent of the particular transmission subsystem and requires only a reliable ordered data stream channel. (Postel, 1992). With this information anyone who can write a computer program, in almost any language, can communicate through the Internet as long as the proper interaction protocol is maintained.

The protocol model greatly increases access because it separates content from the delivery technology. By ignoring the delivery platform, incompatible systems can communicate and share information. This approach is superior to the traditional courseware development model because it increases access, improves flexibility and adaptability as well as reducing cost through more efficiently maintaining the installed technology base. What is needed is a mechanism for developing the appropriate educational protocols.

The open protocol model is at a great disadvantage because the classroom has many distinct instructional paradigms for content delivery such as the lecture, class recitation, or small-group collaborative discussions as well as for student evaluation. Programmers working within the traditional method of courseware development were much better able to manage the sequencing of lesson content because they controlled how a specific user interacted and navigated through the content.

A further problem is with HTML, the underlying encapsulation language. HTML was designed to simply display static document types such as reports or journal articles. It was never designed to structure, order, or define data. "HTML formats how you present a Web page's data and it is not designed to work with what that data represents." (Holzner, 1998, p. xi) Recognizing this discrepancy there has been a call for the development of better markup languages that could represent not only the format but also the structure of the data. This would allow for a much more dynamic and flexible data delivery system.

The World Wide Web Consortium (see W3C) is developing the Extensible Markup Language (XML). XML is similar to HTML in that it is a simple text-based system. However, XML provides a more structured and ordered approach to content creation. XML is "most often used as data-description language allowing us to organize data into data structures - even complex data structures...You can tailor the data as you want it; the most attractive feature of XML is that you can create your own tags." (Holzner, 1998", p.4). This lets an author structure data into appropriate elements "without having to restructure your data to fit a predefined markup language." (Holzner, 1998, p. xi).

Because the markup tags in XML are not fixed ahead of time, as they are in HTML, the language needed a way of informing the browser how to read the document structure. For example in a set of hospital records, the browser would need to know when a patient record ended and a new record started. Further the browser would need to know what type of data defined a patient, such as name, physician's name, room number or the medication they were taking. XML solves this problem through the use of the Document Type Definitions (DTD). The DTD (see Figure 1) defines not only the overall structure of the document, but also individual elements within the document.

```
<?xml encoding="US-ASCII"?>
<!ELEMENT patientRecords (patient)*>
<!ELEMENT patient (name,doctor,room)>
<!ELEMENT name (#PCDATA)>
<!ELEMENT doctor (#PCDATA)>
<!ELEMENT room (#PCDATA)>
```

Figure 1: Sample Document Type Definition

In this example the DTD defines the document's data as a collection of patientRecords. These patient records are a compilation of individual patients that are identified with a name, a doctor and a room number. The DTD then further refines that definition in that all of those elements are typed as PCDATA (parsed character data). A browser would have no problem using this data to correctly display the information . Further, we could easily determine the number of beds currently occupied, patient names, and which physicians were caring for them.

The DTD can be carried by the document so that the browser could properly determine the unique structure of the document. However, the DTD can also be referred to as an external file located on the Internet, which could define a whole class of documents. As long as the documents are properly formatted, a browser could determine how to use them. Consequently, XML is perfectly designed to represent an e. e. cummings poem, as well as, a student record database. Using open standards, schools would be able to maximize their installed technology base while integrating effective and congruent instructional solutions.

Developing the prototype

XML provides a good model for courseware development because it allows content experts, software developers and even classroom teachers access to the same educational material. Because of the DTD, XML provides a mechanism for shaping and disseminating the required protocols to standardize the delivery of instructional content.

In order to explore the implications of how the new open protocol approach to authoring would work we selected a simple multiple-choice exam to prototype and study. This type of exam is used in a variety of educational settings and is generally well understood. Multiple-choice exams are constructed around a statement with a set of possible answers of which only one answer is correct.

We configured a Pentium 166 to run the Apache Web Server 1.3 under Red Hat Linux 6. The Pentium was also configured to use the Blackdown Java port of Version 1.1.7 for Linux. A simple threaded Java Server was authored that processed individual student and exam data and saved it to a file located on the server. In addition, we used the IBM XML Classes 1.1.16 to facilitate the parsing of the exam data within the Java Applets. The browser for all the groups in the study was Microsoft's Internet Explorer 4.5 for the Macintosh. Internet Explorer allows for local access of additional class resources through the Java Preference window.

The Study

XML and Java are new technologies. We were initially interested in investigating the effect of bandwidth on user performance. Secondly we were interested in some basic screen layout issues. We wanted to investigate the effect of item layout on student performance in a multiple-choice test.

The study was designed as a usability study that investigated the issues of screen layout and bandwidth on student performance. Usability testing (Hom, 1999) uses a mixed mode methodology that incorporates both quantitative as well as qualitative methods. Therefore, besides server log data we also video taped each exam session, maintained a session journal and interview students at random.

The first group used a Java Applet that accessed local XML classes through the individual student's browser that used a non-wrapping screen layout as illustrated in Figure 2.

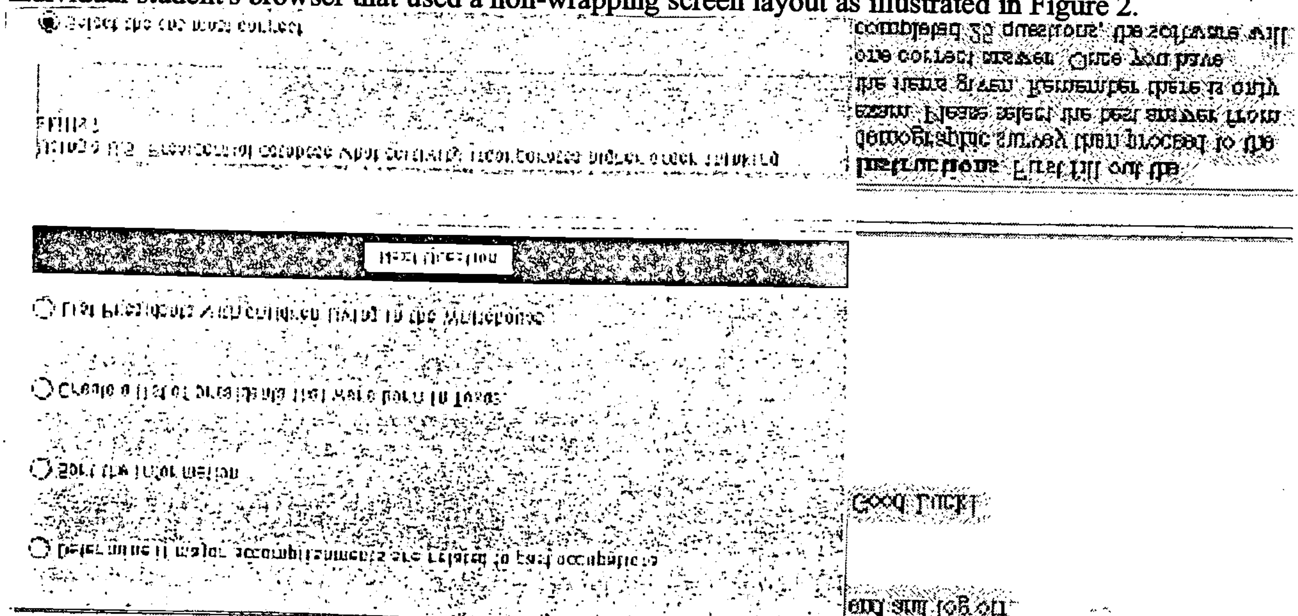

Figure 2: Non-wrapping items

The second group as illustrated in Figure 3, also accessed the set of local classes through the browser. The difference in this group was in the screen layout, which used a centered wrapping text design.

Groups 3 and 4 accessed the required XML classes that were embedded within the Java Applet code. This resulted in an applet that was 17.55 times larger for groups 3 and 4 in contrast to groups 1 and 2 (351 Kilobytes versus 20 Kilobytes). Group 3 used the non-wrapping text interface while group 4 used the centered wrapping text interface.

Participants: The participants for this study were 68 pre-service teachers taking a technology integration course. During the course the students are required to pass a simple multiple-choice test based on primarily factual information from the textbook, class lectures, discussions and software specific skills.

Materials First we created the Document Type Description for the exam as described in Figure 4. XML allows for the definition of new data structures.

```
<!ELEMENT exam (question)*>
<!ELEMENT question (query,items,answer)>
<!ELEMENT query (#PCDATA)>
<!ELEMENT items (a,b,c,d)>
<!ELEMENT a (#PCDATA)>
<!ELEMENT b (#PCDATA)>
<!ELEMENT c (#PCDATA)>
<!ELEMENT d (#PCDATA)>
```

<!ELEMENT answer (#PCDATA)>

Figure 3: Document Type Definition (DTD)

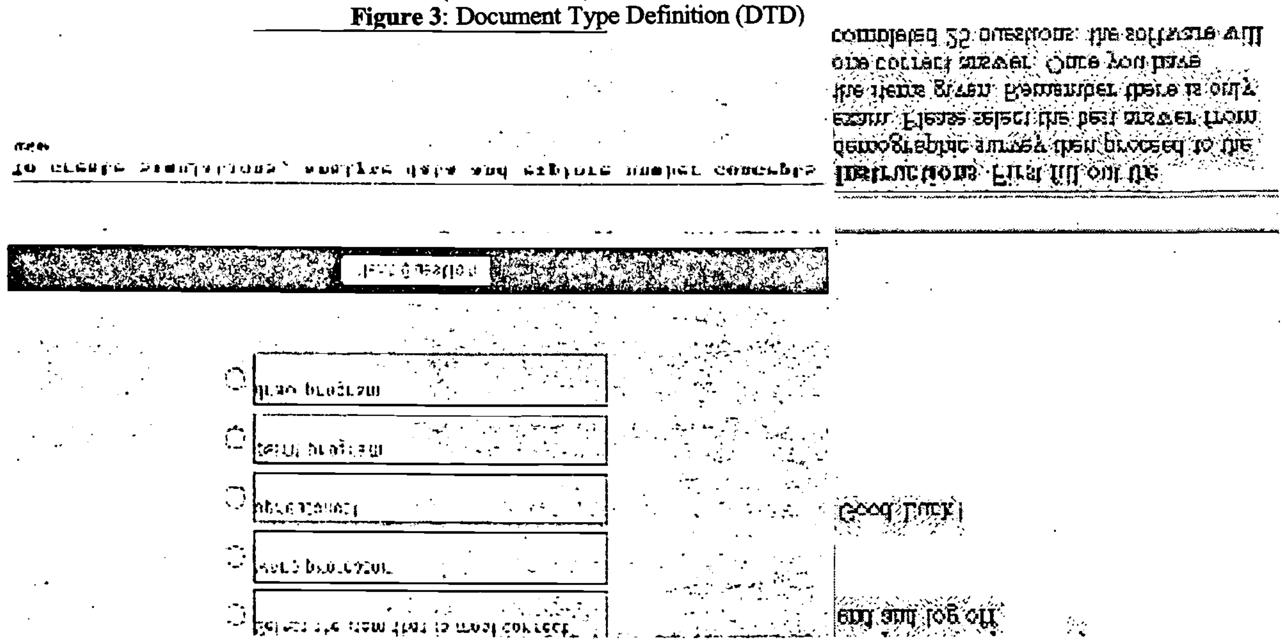

Figure 4: Wrapping item list

The data set was then encoded using the DTD as the framework and saved in a simple text file on the server. We designed a test bank of 25 questions that was based on factual information taken from the textbook, class lectures and discussions. For example, consider the following question:

```
<question>
        <query>One kilobyte is equal to _______ bytes of data.</query>
        <items>
                <a>1000</a>
                <b>1024</b>
                <c>2000</c>
                <d>2224</d>
        </items>
        <answer>a</answer>
</question>
```

Notice how the question is designed. XML requires complete start and end tags, so that each question is required to be marked by <question> and the </question> tag in order to create a well-formed document. Remember that the exam is made up of questions and each question has a query, item and answer elements. As we can see the query is a simple string that makes up the actual question. Next we define the four items from which the student must select the correct item. The answer is provided in the answer tag. This allows the applet to track the individual student performance within the exam.

Procedure

There were 24 sequentially numbered Macintoshes in the computer lab. As a student entered the lab they were asked to sit at the next open computer in sequence starting with computer 01. The instructor proctored the exam and instructed them to open and read a simple web page located locally on their machine. The web page consisted of a set of instructions and acted as a gateway so that they would enter one of the four groups. Half of the machines accessed either the remote or local classes and half used the non-wrapping or wrapping screen layout.

Once the student had read the instructions they continued to the next screen that displayed a simple demographics survey as shown in Figure 5. The students entered their name, the computer's number, their section, age, gender, and estimated internet usage.

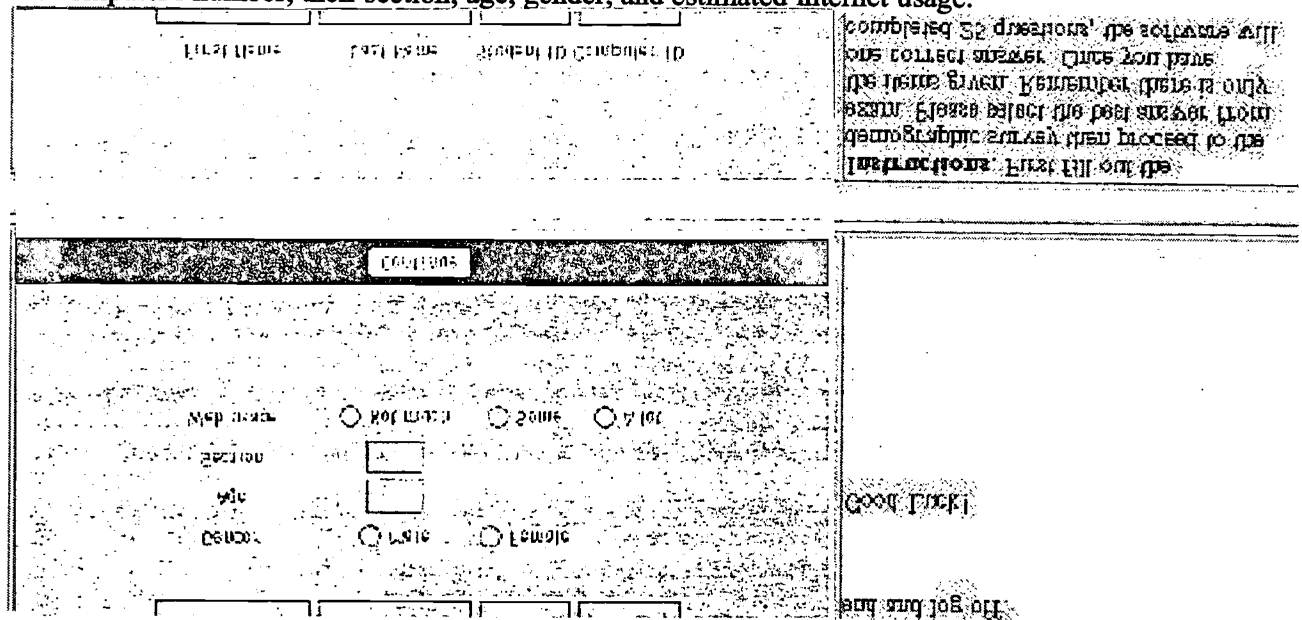

Figure 5: Demographic screen

Once the student had finished answering the survey questions, they began the exam.

Each student was presented with 25 questions in a random sequence. Once they had answered the questions they were presented with their score and then instructed to log off. The data was then transmitted to the server for recording. In addition to the demographic data we recorded the time that the demographic page was displayed, the order of questions, the student's individual responses and the student's total exam score. By identifying each individual computer number we were able to correlate the access time based on the server logs for each student. These were then arrayed with the other data. In addition to the server-based data we also videotaped each exam session, logged any problems that the students may have had. At the conclusion of the exam session we also interviewed a group of randomly selected students.

Results

Results indicate that an open systems approach is a feasible way to design instructional tools. Data encapsulation using XML was easily created, maintained and modified thereby allowing for a seamless interaction on the part of the faculty. The students reported on the exit interviews that taking the exam was easy and they liked the convenience of immediate scores.

One technical problem did surface while using the software. The current version was designed so that the students could not return to a previous question once they had posted their selection. This was an option that students consistently voiced in the exit interview. However, only one student attempted to return to a previous question by clicking on the browser's back button. To her dismay the exam restarted. That was a design flaw in the implementation. One possible solution would be put the exam applet in a separate window and disable the back button.

The location of the required classes was not a factor in the student exam performance. On average the Java Applet took about 5.5 minutes to load once the students accessed the web site. Students who used the remote classes scored, on average, 21. 8 while those that used the local classes had an average score of 22.0. This difference was not statistically significant ($p = .81$).

According to the data, XML is suitable for developers of instructional software because we found no performance difference in using local or remote classes. If you can control the lab situation local classes would be preferable simply to reduce server load whereas if you are delivering the exam over the internet then you can guarantee access by including the required classes. Security still poses a problem, however. The Apache web server supports session tracking so that we can program a more robust authentication scheme for a more secure access.

Educational Implications

The implications for an open protocol-based educational technology system are crucial. First it separates content from the delivery technology which allows the development of appropriate educational material regardless of the available delivery technology. Developers would be free to concentrate on effective and appropriate instructional materials targeted to what teachers need for the classroom. Further it would provide teachers with the ability to construct, modify or share computer-based materials for their classrooms. This would reduce the time needed to create course materials thereby substantially lowering costs. Much work needs to be done, because the Internet was developed to support science and industry. Issues of message structure, educational protocols and effective content handlers must be developed before the real transformation takes hold. These items keyed the rapid rise of the Internet for business and science because they created portable sources of information. The tool you choose to complete a job is intimately tied to your success or failure. Matching user skill with the correct tool creates an optimal situation guaranteeing the most effective, efficient and elegant solution. (Simon, 1996). If education is to emulate the growth of the Internet, there is a need for similar tools that truly match the needs of instruction.

References

Berger, R. (1999). Microscared: The challenge of open source software. *Linux Magazine, 1* (1), 25-29.

Blackdown Organization (1997). Linux Java 1.1.7. [Available online at:] http://www.blackdown.org.

Holzner, S. (1998). *XML Complete*. New York: McGraw-Hill.

Hom, J. (1999) The usability toolbox. [Available online at:] http://www.best.com/~jthom/usability/

International Business Machines (1998). XML Parser for Java. Available online at http://www.alphaworks.ibm.com

Metrowerks (1999). CodeWarrior Professional Release 4. Austin: TX: Metrowerks.

Postel, J. (1982). *Simple Mail Transport Protocol* (No. RFC 821). Information Sciences Institute, University of Southern California.

RedHat (1999) RedHat Linux Version 6. [Available online at:] http://www.redhat.com

Simon, H. A. (1996). *The science of the artifical* (Third ed.). MIT Press.

The Apache Software Foundation (1999). Apache server 1.3. [Available online at] http://www.apache.org

W3C: The World Wide Web Consortium (1999). Extensible Markup Language (XML). [Available online at:] http://www.w3.org/XML/

NOTICE

Reproduction Basis

☒ This document is covered by a signed "Reproduction Release (Blanket)" form (on file within the ERIC system), encompassing all or classes of documents from its source organization and, therefore, does not require a "Specific Document" Release form.

☐ This document is Federally-funded, or carries its own permission to reproduce, or is otherwise in the public domain and, therefore, may be reproduced by ERIC without a signed Reproduction Release form (either "Specific Document" or "Blanket").

EFF-089 (3/2000)