Promoting Web Applications in the IT Educational Environment

Boris Bacic
Computing - Systems & Technology Group
Auckland University of Technology
Auckland, New Zealand
boris.bacic@aut.ac.nz

ABSTRACT

In the present environment in which Internet applications are growing in importance IT educators should aim to include in their teaching practice the design of Web applications as a powerful tool to enhance the teaching/learning process. The main focus is in innovation in computing practice and education.

KEYWORDS

Web design, Web development, On-line education, On-line learning.

1. INTRODUCTION

The aim of this paper is to help and encourage motivated educators who:

- Wish to start Web server development;
- Know at least one programming language; and
- Like learning by doing.

This article will also reveal ideas and recommendations from some of the past and present experiments using local Intranet applications

2. PROJECTS AND IDEAS

In the educational environment as in business and e-commerce there are many Web applications that could be done to improve efficiency, reduce the administration overhead and facilitate on-line learning.

Common examples of Web applications that are often found in education are:

2.1 Distance Education Courses

Shifting a course on-line should be done by a team of professionals. A possible area of experimenting in the new technology would be to try to split on-line courses into two parts or layers. The first part would be a portable, platform independent text based (Extensible Markup Language) XML - database. In the second part, which would serve to produce the front-end we would use another text-based standard called XSL (Extensible Stylesheet Language). The benefits of using the layered approach where we

separate the contents from its presentation are that the whole system is easier to maintain and modify. Instead of converting MS Word documents to (often "extra-large") HTML files, we should plan to build the system that supports an XML database.

2.2 Computer Based Training (CBT)

Major benefits of in-house developed CBT would be that:

- The authors are likely to be more available for support and further modifications;
- The content is inline with current curriculum; and
- An opportunity for IT educators to become familiar with the latest techniques of Web design.

Major benefits of bought CBT would be that:

- The software is developed and tested by professionals who make educational software;
- The presentation and navigation part is likely to be better than the in-house developed CBT; and
- Cost. Students may be in favour of using some of the popular CBT courses.

2.3 On-line Examinations

Advantages of on-line vs. theory examinations:

- Less administrative overhead especially for large groups of students;
- Quicker response time with less effort for educators to prepare student's feedback; and
- Inconsistency of marking across the group of students is reduced to minimum.

Disadvantages on-line examinations

- Limited freedom of exam questions' content (eg. multichoice rather than descriptive answers);
- Increased stress level for students, greater number of different problems could be introduced. Possibility that the exam may not be finished within the planned time frame;
- For small number of students who do the resits all facilities including the system support and specialist/supervisor must be available (again).

The system must support the following business rules:

- It must positively identify each participant;
- Each participant must be able to contribute without interfering with others;
- A participant should not be able to see other participants' submitted work; and
- A participant can use the system and submit his/her work between the start and end time.

Optional rules:

- A participant may or may not be able to see his/her own submitted work;
- A participant may be able to submit his/her own work more than once;
- Full or partial feedback on submitted works may be immediate; and
- Feedback may be given through web browser, e-mail or network message.

2.4 Submission of Assignments

Typical common solutions are:

- E-mail attachment
  - Pros: simple and efficient way to reach intended recipient from home or from within the institution.
  - Cons: It requires extra manual intervention for saving work.

- File Transfer Protocol (FTP). Full automation can be achieved. A sender may compress his/her work preserving the original directory structure. Automated process (eg. Unix daemon) may expand the compressed file to the subdirectory with student's id as a name. Business rules can be implemented through file/directory rights.
- Standard EXE file or Active X control. Full automation can be achieved through custom-made program that can:
  - Submit hidden information about the sender that could not be achieved using standard HTML files loaded from the Web server.
  - Store directly to the common directory or
  - Dynamically generate HTML document and load it into the Web browser (like on-line registration) for submitting purposes.

2.5 Automated Marking and Statistics

Require programming effort, knowledge of CGI and optional database application experience.

3. TECHNOLOGIES

To build the local Intranet we need at least one computer with TCP/IP protocol and Web server installed. On a single machine it is possible to have many Web servers installed, but only one can be active at one time at a particular TCP/IP port (NB: default port is 80. In case of using another port we should also specify it in URL, eg. local web server using port 8001 should be accessed through http://localhost:8001/). For testing and implementing a Web server application we need a computer lab where many computers connected into the network using TCP/IP protocol can access a computer running a Web server.

4. BUILDING THE HTML FRONT-END

Before choosing the favorite HTML editor we should compare the few HTML editors based on the following benchmarks (beginner/intermediate to advanced use):

GUI - user friendliness, ease of use eg. working with tables, hyperlinks (including e-mail, and links to CGI), creating and formatting input forms for CGI; Can HTML code be previewed using external browser (eg. IE and NN), is it a "two-way tool" (GUI and HTML text editing/previewing/WYSIWYG allowed); Insertion and formatting graphics; Themes; Image editing capabilities; Possibility of insertion of Java applets, JSP, ASP, Shockwave or Flash; Possibility to retrieve pages from the Internet; Can we create automated JavaScript (eg. rollover buttons); Working with Frames; Site Map view and site management; Automatically Update links; Publish to server, multi-user mode; Clean code; Documentation and Help;

Processing at client's front end can be done using Java and Java applets, Jscript or VBScript.

Limitations: To satisfy security and privacy most of the technologies that serve to build web pages do not have the functionality of standard executables of Microsoft's ActiveX technology.

Free editors: MS FrontPage Express, Netscape Composer (part of NN)

Mid price range: Namo WebEditor, MS FrontPage 98/2000

Top price: Macromedia DreamWeaver

5. BUILDING THE BACK-END

To build complex Web server applications in this article we will focus on CGI (Common Gateway Interface), which was the first dynamic content technology in the Internet era. A major benefit for using CGI technology is that CGI applications are relatively easy to create using a broad array of development languages that could run on any platform. When a CGI application is requested from a client's browser (eg. http://www.mysite.com/cgi-bin/CGIdemo.exe/) the Web server spawns the CGI application, whose standard output is redirected as a reply to the client's browser. The biggest drawback of the CGI protocol is that every Web request runs an executable that typically resides on a hard disk. Other popular server-side technologies details of which are worth searching for the Web are: Java Servelets, JSP, PHP, ASP, ISAPI and non-free technologies like: Macromedia's UltraDev, Allaire's ColdFusion.

Another important factor when choosing a language for CGI is the ease of database access and manipulation.

6. BUILDING THE PROJECT TEAM

In educational institutions building a team of educators with different areas of expertise means that they could cover a broad area of technology, and introduce new ideas and frameworks. Another important aspect is that one could help others to gain knowledge better than in the team where only one or few technologies are used. To achieve efficiency and reduce "time to market" - as an analogy in education the project leader should test the proof of concept, reject it or accept it and could delegate/involve others to finish different parts of the project and also gain

	Type of information	Access to Database	Additional logic required	Examples
1	Simple information	No	No	Static Web pages, notifications main page, on line courses
2	Cataloging	Yes	No	Generated Staff Web Pages
3	On-line activity	Yes	Yes	Exercises, On-line marking/examining

Table 7.1: Type of Web information

the new knowledge/experience.

7. RECOMMENDATIONS FOR DEVELOPERS

Common approaches to the problem described in Table 7.1 and alternative ways of solving them:

1 Simple Information: Content written in MS Word converted to static HTML. If possible use content written (or partially pasted from clipboard) into HTML editor.
2 Catalogue: Database holds HTML constructs "building blocks" and data. Using scripting we could produce HTML files as an output. If possible use: XML and XSL.
3 On-line Activity: Build the Web server application combining the HTML templates and technology proprietary tags when extra logic is required.

8. PRACTICAL EXAMPLES - CASE STUDIES

A group of distance education courses (eg. IN200) at AUT (running on Apache for Linux). Static pages were developed and maintained using some of the popular HTML authoring tools. To separate content from the presentation in the near future the courses will be ported to XML/XSL technology.

Generator for staff web pages that could be expanded to offer flexibility of information on-line. To build a proof of concept we used an Access database that contains the front end for data entry, a database and "building blocks" - HTML constructs stored in the database as well. To achieve additional programming flexibility Java has been used as well. At present we plan to use JSP (Java Server Pages) with the Tomcat Web server. The database part is exported into XML where XSL is used for presentation layer.

ON-LINE QUIZ: Number systems lab exercise. Students are required to submit small number of on-line test like exercises. To reduce administrative overhead, and provide better feedback, we will use Delphi to develop the Web server application. A database will contain the pool of questions and answers, including dynamic parts of HTML information. Using a technique similar to JSP or Cold Fusion tags, the Web server will generate the pages through CGI protocol, maintaining "the state" of students who are using the system. A Web application should also be responsible for generating the individual feedback to students and updating the database about their progress. Delphi code will be recompiled using the latest Kylix (Delphi for Linux) and as a database it will use Interbase or MySQL.

In all three systems mentioned above HTML output is optimised for speed. To add extra features to HTML output all three systems could be modified by just changing their templates.

9. CONCLUSION

Educators know many different ways of learning. In general educators as self-learners are more interested to learn the principles or what is possible to achieve using a new technology.

Suggested steps for self-learners:

1. Obtain the software:

- Free Web Servers (Apache, JigSaw, Tomcat or Microsoft's IIS or PWS)
- HTML editors of your choice (including all free and shareware products and some trials)

- Multipurpose text editors (eg. Med editor http://www.utopia-planitia.de/us/downloadwin.html)
- Free image viewers (IrfanVewer, ACD systems, etc.
- Image editors (to be able to edit/export GIFs and JPEGs at least).

2. Study the Technology

- (HTML, CGI protocol, XML/XSL etc.) and develop small examples using the local Web servers.

3. Choose your

- Directions: eg. Multimedia, Java, Databases and how to interface them.
- Projects and Team.

ACKNOWLEDGMENTS

I wish to express my appreciation for support given (without imparting responsibility) by Ewing Caldwell, Brian Cusack, Doug MacKenzie and Ronald Zucker as teaching colleagues and developers for the support they gave me in this paper.

BIBLIOGRAPHY

Bacic, B. (2000), "Web Design Techniques using Delphi". Proceeding of the NACCQ Conference, Wellington, p.419.

Calvert, C. (1999) "Delphi 4 Unleashed". SAMS Borland Press, ISBN:0-672-31285-9.

Graham, Ian S. (2000), "XHTML 1.0 Web Development Sourcebook". Wiley, ISBN: 0-471-37486-5.

STANDARDS, PROTOCOLS AND GENERAL REFERENCES

www.w3schools.com/default.asp/, www.w3.org/, www.w3.org/XML/, www.startkabel.nl/k/xml/, devcentre.org/research/webarchitecture.htm

Web servers downloads

www.apache.org/

Scripting: msdn.microsoft.com/scripting/

HTML editing: home.netscape.com/ www.namo.co.nz/, www.namo.com, www.macromedia.com/software/dreamweaver/

Managing XML: www.microsoft.com/biztalk/, www.softwareag.com/tamino/

Server side technologies: www.macromedia.com/software/ultradev, www.coldfusion.com/Products/coldfusion

On-line quiz: web.uvic.ca/hrd/hotpot/wintutor/index.htm

Free-downloads: www.tucows.co.nz/, www.tucows.com/, home.netscape.com/computing/download/index.html