

# THE WORLD WIDE WEB AND INSTRUCTION

**Dr. Ann E. Barron, Associate Professor**  
**University of South Florida**  
**Tampa, Florida 33620**

**Brendan Tompkins, Programmer/Webmaster**  
**Analysis & Technology**  
**Orlando, Florida 32826**

## ABSTRACT

The Internet and the World Wide Web provide great avenues for the dissemination of training materials. Of special interest to trainers who are exploring new instructional paradigms is the potential for worldwide, on-demand instruction, such as interactive tutorials, simulations, and testing.

This paper explores the readiness of the Internet as an instructional delivery environment capable of the dissemination of multimedia-based materials that provide interactive transactions with a learner. The advantages and disadvantages of techniques that represent opportunities to provide meaningful interactions will be outlined. In addition, design guidelines will be provided for creating instructional materials on the World Wide Web. These guidelines include recommendations for page layout, text styles, graphics formats, multimedia components, and navigation options.

## BIOGRAPHIES

Dr. Ann Barron is a frequent presenter at national and international conferences, including SALT, AECT, NECC, AERA, and ASTD. She has published five books, including *Multimedia Technologies for Training: An Introduction*, and she is the Executive Editor of the *Journal of Interactive Instruction Delivery*. Ann is currently an Associate Professor of Instructional Technology at the University of South Florida and the Director of the Florida Center for Instructional Technology. She may be reached at [barrona@mail.firn.edu](mailto:barrona@mail.firn.edu).

Brendan Tompkins is an Internet Development Specialist for Analysis & Technology, Orlando, Florida. He has created numerous documents for the World Wide Web. Brendan earned his Masters degree in Instructional Technology from the University of South Florida. He may be reached at [tompkins@Mercury.or.atinc.com](mailto:tompkins@Mercury.or.atinc.com).

# THE WORLD WIDE WEB AND INSTRUCTION

## INTRODUCTION

The Internet currently stores a great deal of information related to training materials. Static materials, such as class notes, student handouts, and copies of instructors' presentations are all easily disseminated via e-mail or file transfer protocols. Of special interest to educators and trainers exploring new instructional paradigms is the delivery of interactive, on-demand instruction. This paper explores the use of the Internet and the World Wide Web as instructional delivery environments, and it provides guidelines for developing Web pages.

## ONLINE COURSE DELIVERY

When the Web was originally conceived, it provided hypertext documents with links to graphics, sounds, and video files. The interactivity was limited to navigating from one document or file to another. Only very recently, with the development of Web-savvy programming tools like Java and Shockwave, can it be said that the potential for interactivity on the Web are equal to those of a stand-alone computer. This increase in capabilities has strong implications for instruction and training on the Web, but to fully understand the implications, it may be helpful to classify Web-based instruction.

One way to classify the instruction delivered through the Web is to focus on the amount of interactivity and the computational requirements that are needed to create the courseware. Using these criteria, Web-based instruction can be divided into three categories. Each of these categories allows an increasingly complex interaction to take place; Type I is easier to develop and much less interactive than Type III.

Type I information is sent from a remote computer (server) to a browser on a local computer (client) which formats and displays the data. Type II interactivity is based on a request from a client computer, additional programs are executed on the remote server to generate unique, tailored responses. Type III includes computer programs that are passed to the client computer from the remote server. The client computer has extensions which allow the programs to be executed locally.

### Type I

Type I instruction uses the standard features of Web browsers, such as Netscape, to display hypertext documents with links to images, sound, and video. With Type I instruction, the user can be guided through a traditional Web page with sections of text, graphics, and links. It is also possible to present students with simple multiple-choice interactions by creating hyperlinks to pages with appropriate feedback. There are many examples of this type of instruction on the Web. One, "Interactive Technologies," was created by the Instructional Technology program at the University of South Florida. Created as a resource for K-12 educators in the state of Florida, this tutorial contains information about the videodiscs, CD-ROM, Photo-CD, and CD-i. This program can be accessed at [http://www.coedu.usf.edu/inst\\_tech/publications/NewTech/](http://www.coedu.usf.edu/inst_tech/publications/NewTech/). Another example of Type I instruction is "Anatomy of an Eye" which is available at [http://www.netscape.com/comprod/products/navigator/version\\_2.0/frames/eye/index.html](http://www.netscape.com/comprod/products/navigator/version_2.0/frames/eye/index.html). This program incorporates HTML frames to provide a feedback and branching on the same screen.

Type I instruction can be created by using the Hypertext Markup Language (HTML). HTML is an ASCII-based language that specifies the layout of text, graphics and the functionality of hyperlinks. Using HTML, links and feedback can be included that will branch the student to another point on a page, a different page, or a multimedia component.

### Type II

To create more sophisticated interactions on the Web, at least Type II instruction must be employed. Here, the server runs a program via a method called the Common Gateway Interface (CGI) and sends unique information depending on a response the user has made. Possible features in Type II programs include database lookup and searching, clickable areas (hotspots) on images, evaluative feedback, and the creation of dynamic Web documents. In a program, created by David Tai at the University of South Florida, forms are used to accept user inputs and generate feedback. His program, "HTML Tutorial," can be accessed at <http://fcit.coedu.usf.edu/publications/html/>

Another course utilizing Type II instruction is also being developed at the University of South Florida. As part of a program about installing a school network, the user interacts with a Jeopardy-type game called the "Puzzle Peril." Here, the student is presented with true-false, multiple-choice, and fill-in-the-blank questions. Each question is generated dynamically by the server's CGI which reads the question and correct answer from a database. Then, a separate program evaluates the student's response, provides feedback, and tallies the student's score. This all combines to create very sophisticated interaction, close to the type commonly seen in traditional off-line computer-based instruction. Puzzle Peril can be accessed at <http://www.coedu.usf.edu/fcit/game/>.

Type II instruction relies on CGI programs that are written in a computer language such as C or Perl. Some CGI programs can be obtained and used without requiring the author to learn to program. This is the case when using server-side image maps to create hot spots on images. More often, however, these programs must be created for each individual application. This usually involves familiarity with the server's operating system, knowledge of programming, and a great deal of time spent programming and troubleshooting.

Although Type II instruction can provide rich interaction types, there are additional limitations. For example, Type II instruction may slow server systems down due to extra computational strain. Also, many service providers limit the extent to which Web authors can utilize CGI-based applications. If you are planning to use this type, consult your provider or system administrator.

### **Type III**

Type III instruction is the most complicated, and also has the most potential for delivery of instruction via the Web. Here, a program is written (and sometimes compiled) in a form that can be sent over the Internet. This form differs from normal executable code in that it is usually platform-independent, and it contains additional Internet links and security features. This allows the same program to be run on a computer running Windows 95, Macintosh OS, or UNIX. Because of the security features, the user can also be fairly certain that the program does not contain malicious code or viruses. Security and platform independence are key advantages to this type of instruction that make it a viable means for distributing programs on the Web. Two methods of employing Type III instruction are Java and Shockwave for Director.

Java is a C++ based language that has recently garnered a great deal of attention. With Java, the programmer creates small, compiled programs called "applets" which are sent over the Internet and can be incorporated into Web pages. Using Java, one could conceivably create any type of program that can be run on a computer -- including word processing programs and fully interactive CBI. The possibilities are limitless. To see an example of a Java application which allows the user to build and test digital circuits, access <http://www.lookup.com/Homepages/96457/digsim/load.html>.

Shockwave for Director is an extension of Macromedia's Director program. Director is a time-based multimedia authoring system with a rich programming language called Lingo. Director can be used to create traditional CBI, animations, games, and other multimedia applications. Macromedia has developed a method of converting Director programs (called "Movies") into a form that is cross-platform (Windows and Macintosh), secure, and contains Internet extensions. An example Shockwave applications can be seen at <http://www.sageinteractive.com>.

Java, and to a lesser extent, Shockwave, require a great deal of programming knowledge and skill to create applications that can be used in an instructional context. However, as Java develops, authoring environments which make the task of creating Java programs easier are expected to enter the market. When this happens, it will bring the creation of Java applications into the realm of most mid-level programmers. Director programming is already within the reach of most programmers. In addition, Internet-ready versions of other authoring systems and programming languages are expected to develop, and this will surely increase the amount of Type III instruction on the Internet.

## **DESIGN GUIDELINES**

Because it is still in its infancy, few design guidelines exist for delivering training on the Web. A major constraint for all three types of Web-based training is the bandwidth available for transferring the Internet files. Although applications such as Shockwave or CGI programs can provide high-levels of interactivity, they will be impacted by the type of conduit that is being used to transfer the files. For example, if a 500K Shockwave file is being

transferred through a modem, it will take substantially more time than if it is transferred through a T1 line with a direct connection. In either case, the response time will probably not be as fast as it would be from a CD-ROM or hard drive.

The following guidelines are proposed for the development of Web pages, in particular those that will be used for training. The guidelines are based on personal experience, Web design documents, and reference books. The guidelines should be interpreted broadly, based on specific content, the intended audience, and the format of the data.

## **General Guidelines**

*Analyze your target audience.* Before designing Web-based instruction, you must be familiar with the characteristics of the target audience. In addition to determining the content of the course, this analysis can help to decide on whether to employ Type I, Type II, or Type III instruction. For example, Type III instruction is more appropriate if there is a very fast Internet connection to all of the students. Also, Type II instruction is required if management records and student tracking is important.

*Determine your content.* Your Web documents will be most effective if you analyze the content carefully before you begin. After the knowledge and skills that you intend to teach are outlined, you can best determine the amount of interactivity that will be required and the design strategy to employ.

*Carefully title your Web pages.* A general CBT courseware design guideline has always been to title all of the screens. This practice is even more essential with Web-based training, where students will have a great deal of control over the navigation through the course.

*Provide a method for feedback.* It is quite easy to include an e-mail feedback method for users -- either by a MAILTO tag or a form that elicits typed input. Through this device, you can solicit information from the students and help to address any confusion or questions they may have.

*Design for all different browsers.* Just because you are using the latest version of Netscape, it does not mean that everyone else will be using the same interface. Numerous browsers are available, and it is an easy task to test your pages on several different ones before releasing the pages. In particular, it is important to consider the text-based browsers (such as Lynx) and to include alternate text for all graphics.

*Design for the default page size.* One frustrating element of designing for the World Wide Web is that you do not have control over the delivery environment. The user can resize the window, which may alter the amount of text on each line. In general, it is best to design for the default screen size and anticipate that some users may resize the screen and make it smaller or larger.

*Pages should not be longer than 2-3 screens.* Long Web pages cause users to lose their orientation. It is best to divide the content into chunks of 2-3 screens of material, with links to additional documents. Many designers also recommend that the first (home) page should be only one screen length, which does not require any scrolling by the user.

*Provide separate files for printing long documents.* The disadvantage of short pages is that if users want a printout, they must print several documents and "piece" them together. To address this issue, it is wise to provide a link to a separate file (with the entire text) that can be printed or saved as one document.

*Always pilot test your Web pages.* Before releasing your Web pages to the rest of the world, conduct a pilot test; watch as a sample of the target audience navigates through the pages, and make revisions if they are necessary.

## **Graphics**

*Make sure the graphics are relevant and necessary to convey the message.* With delivery of multimedia on CD-ROM or other mass storage devices, the amount of size of graphics is not generally an issue. However, on the Internet, each picture must be transferred separately from the remote computer. The transfer time can vary based on the type of connection (a modem vs. a direct connection) and the amount of traffic on the lines. Pictures, such as a

graphic of your cat or dog, may cause frustration for a user who has to wait 2 or 3 minutes for it to transfer and display. Before incorporating a graphic, assess its pertinence to the overall message.

*Limit the size of graphic files.* Graphic files should be kept as small as possible, with no one graphic being larger than 50K. This limit may mean that you use a banner rather than a full-screen graphic, or that you create the black and white graphics instead of using 256 colors.

*Avoid using large graphic menus.* Most users grow impatient when waiting for large graphic files to display. Transmission rates for large graphics is a particular problem for users with slower modems. If the wait is over 20 seconds, it is likely that the user may lose patience, terminate the transmission, and select a different site (Lynch, 1995; Shneiderman, 1992). Home pages with large graphics will always load more slowly than text-based pages or pages that incorporate smaller graphics and typographic styling for visual impact (Lynch, 1995).

*Limit the number of graphics on each page.* If a large number of graphics are on the same page, the transfer time will be increased, perhaps causing the user to "Stop" the transmission and look elsewhere.

*Limit the width of graphics to less than 472 pixels.* Most browsers default to about 500 horizontal pixels on Macintosh and Windows computers. To ensure that the user will be able to see the entire graphic, limit them to 472 pixels in width.

*Use interlaced (progressive) graphics wherever possible.* If graphics are interlaced, they appear progressively on the screen -- first a very-low resolution graphic appears, and it gradually becomes more clear. Interlaced graphics provide almost immediate feedback for the user about the nature of the graphics, and they allow much faster access to the text and links on the page.

*Include width and height information in the HTML code for the graphics.* Graphics will load much faster with some browsers if the width and height are specified (<IMG SRC="Sample.GIF" WIDTH="400" HEIGHT="150">).

*Use JPEG compression for photographs and other continuous tone images.* Graphics on the World Wide Web are generally stored in either JPEG or GIF format. If your images are photographs or full-color images, it is best to save them as JPEG images.

*Use GIF compression for line drawings.* The GIF format is best when there are fewer colors or large areas of a single color, such as in line drawings or computer screens.

## **Text**

*If possible, use logical rather than physical styles.* It is generally advisable to use <strong> or <em> instead of <b> or <i>. Some browsers do not recognize bold or italics, but will recognize strong or emphasized text.

*Make sure there is high contrast between text and backgrounds.* Some of the graphical and colored backgrounds make it extremely difficult to read the text. Use dark text on a light background or light text on a dark background.

*Try to limit the length of text lines.* Long lines of text can be hard to read. Try a few techniques to decrease the length of some of the lines. For example, you can place text along side a graphic, or you can use the BLOCKQUOTE tag.

## **Links**

*Use descriptive words for links.* Try to use words other than click [here](#) as links in your documents. In other words, instead of using "Click [here](#) to see a movie" use "Quicktime [movie](#) of two dogs playing." Also, do not use single characters for links -- they may be hard to click on with the mouse.

*Check your links regularly.* The Web is very dynamic. If you are including links to other sites, check them regularly to avoid sending your users to error messages.

*Include fixed links to provide a structure to your Web pages.* The BACK and FORWARD buttons on the top of the screen in browsers like Netscape many link to another site in another country, based completely on the user's

previous path. If you provide a PREVIOUS PAGE, NEXT PAGE, and MAIN MENU on your pages, you will benefit the users by giving them the interface tools to navigate through the information at your site, in the sequence that you intended. These links can be text or icons; if you use icons, make sure that the icons are easily understood by all cultures.

*Include a link to a local home page on every Web document.* Connect every page at your site to a Main Menu or submenu -- otherwise they will become dead-ends, with no choices for the user other than to select a new URL or back out to the previous path.

*Provide information for links that involve large file transfers.* If the link involves a file transfer, provide information about how large the file is and the format that it is stored in.

## **Digital Audio and Video**

*Use only where appropriate.* Because of the transfer time required for digital audio and video files, they should only be included where appropriate and relevant.

*Store multimedia elements in standard formats.* In order to make the multimedia elements accessible to all platforms (Macintosh, Windows, and UNIX), store them in standard formats for the World Wide Web. In most cases, this means the .AU format for audio and the .MOV format for video.

*Minimize the size of audio files.* Whenever possible, compact audio files by using a lower sampling rate, recording in mono instead of stereo, and using 8 bits of resolution instead of 16.

*Keep video files as small as possible.* Video files can be minimized by reducing the frames per second, reducing the frame size, and limiting the number of colors.

*Include information about file size in links.* Provide all audio and video links with information about the file size and format. In some cases, you may include options for different levels of quality and size of files (such as a movie in 1/4 screen size or 1/2 screen size), and let the users determine how long they are willing to wait for the movie to transfer.

## **CONCLUSION**

The World Wide Web offers many advantages for the delivery of interactive instruction. Although the original Internet-based instruction was text-based, there are now at least three types of interaction that can be delivered through the Web. With Type I instruction, information is sent from a remote computer to a local computer's Web client which formats and displays the data. Type II instruction contains information that is generated by the server as the client's request. Type III is the most complex; compiled programs are passed to the Web client that is able to run the code.

The design of instruction on the Web should follow the same guidelines as designing informational Web pages, with special attention to an appropriate amount of student interaction. It is especially important to analyze the content, transfer large files only when they are relevant to the instruction, and provide ways for the student to easily communicate with the instructor.

## **REFERENCES**

- Barron, A. E., Tompkins, B., & Tai, D. (1996). Design Guidelines for the World Wide Web. *Journal of Interactive Instruction Development*. 8(3), 13-17.
- Comber, T. (1995). *Building Usable Web Pages: An HCI Perspective*.  
<http://www.scu.edu.au/ausweb95/papers/hypertext/comber/> Southern Cross University.
- Glover, J. M. (1995). *Top Ten Ways to Tell if You Have a Sucky Home Page*.  
<http://www.winternet.com/~jmg/TopTen.html>
- Holden, G. (1995). *Publishing on the World Wide Web*. Indianapolis, IN: Hayden Books.

- Ives, B. (1995). *Development Guidelines for WWW Pages*. <http://sun.cis.smu.edu/>
- Lynch, P. J. (1995). *Web Style Manual*. Yale Center for Advanced Instructional Media.  
[http://info.med.yale.edu/caim/M\\_Resources.HTML](http://info.med.yale.edu/caim/M_Resources.HTML)
- Nielsen, J. (1995). *Interface Design for Sun's WWW Site*. <http://www.sun.com/sun-on-net/uidesign/>
- Shneiderman, B. (1992). *Designing the user interface: Strategies for effective human -computer interaction*. Reading, MA: Addison-Wesley.
- Tittel, E., & James, S. (1995). *HTML for dummies*. Forest City, CA: IDG Books Worldwide, Inc.
- Top 10 Things NOT to do on a Web Page*. <http://ee.stanford.edu/eecns/www.donts.html>
- Wilson, S. (1995). *World Wide Web Design Guide*. Indianapolis, IN: Hayden Books.
- Wolf, S. J. (1995). *CSUBIOWEB: Designing Web Pages*. <http://arnica.csustan.edu/design.html>