Teachtechnology: Web-Based Instruction's Dual Environment

Among the major incentives for doing research is the exploration of new ideas and contributions to the knowledge base in the discipline. In theory, the reason for doing this is to avoid repeating errors and move on to making new ones. As we develop new theories and test new hypotheses though, we often discover that we have more questions after testing the hypotheses than before.

This article examines some of the pitfalls and barriers of the instructional design process in creating complex courses for the World Wide Web (WWW), and, in particular, the instructional design, development, and delivery of the on-line instructional component, hence the title word "teachtechnology" or teaching with technology. At first, it seems relatively simple to create engaging instruction by taking existing content text and converting it to HTML, then viewing it in a browser (according to Beer (2000), a bad idea). The challenge in this comes from creating and controlling the various environments surrounding the instruction while providing effective learning.

When our research in education must be carried out on living learners, it is imperative that we think through all the possible negative as well as the positive effects of our hypothesis, and eliminate or reduce as many as possible. Our goal is to enhance learning, not inhibit it. We study the effects on learning of color (Dwyer, 1987), fonts, backgrounds, graphics, animation, (Williams, 2000) and screen layout. These elements represent, however, only half of the learner’s environment. We spend a great deal of time analyzing and designing and revising the way we construct the learning environment(Kristof & Satran, 1995), but in most cases we do not take enough time to analyze, design, and revise the other environment, the technical environment, we have created for these learners.

Previously with computer-based training (CBT) or technology-based training (TBT), the technical environment was mainly within the control of the programmer, since the instructional program was loaded on an individual machine. One had more control over the operation, settings, and response of the particular machine. Web-based Training (WBT), however, is not machine- but server-based and gives the learner much more freedom to choose fonts and colors, as well as a host
of other parameters. If they choose to have their settings override those of the author, then there is
the risk that the desired effect will not be achieved.

Designers also ask the learner to look behind the screen, inquiring about their operating system,
available RAM, availability of "helper apps" and plug-ins, and monitor resolution settings.
Surprisingly, many learners are uncertain what operating system they are using or even how to find
out. A number of those wishing to learn online are using a computer set up by someone else and
are uncertain about procedures to change common computer settings and preferences. They may be
unable or unwilling to explore and make an attempt at trial-and-error changes.

We ask them to become instant computer experts when we use any component that is out of the
ordinary. For some, even a simple Java applet in web-based training (WBT) may crash their
browser and we often give them no clue what may have happened or how to correct it, which they
must do before they can continue within the learning environment.

Learners who avail themselves of WBT have computer skills that range from beginner to expert.
Which end of the spectrum should designers aim for or should they skew the requirements? WBT
learners may or may not have knowledge of their computer's capabilities and limitations, their
browser's capabilities, availability of supporting software, or navigational conventions of WBT.
Learners come to TBT/WBT for content instruction and generally, unlike computer games, do not
select a course or tutorial based on technology challenge levels offered.

**More Than Just Content**

There are primarily two environments that an instructional Web designer must consider when
creating WBT instructional units. First, is the learning environment (i.e., the content itself and
supporting structures) and second is the technical environment a designer creates as a consequence
of the content and the content support materials. While it is understood that it is the instructional or
learning environment that determines the final effectiveness of on-line instruction, it is certainly the
second (i.e., the technical environment) that determines the attitude toward that instruction and
toward the learning environment itself.
Schweir (2001), speaking of learning communities, states that "...community members need to feel that they are in a hospitable environment generally, or they will limit their participation in a group setting" (p.6). Likewise, learners attempting to take a course on-line need to feel that they are entering into a technologically friendly experience.

Similarly, Gerson (2001) feels we don't pay enough attention to the psychological needs of learners. Speaking of performance improvement, Gerson says that the learner's "mental state definitely affects how well they perform and whether they achieve their performance goals... each successful performance increases their feelings of competence, confidence, and self-worth." (p.6) Yet we force them into technologically unfamiliar experiences with often unfriendly results.

On-line course instructors know how to utilize both synchronous and asynchronous communication to increase learner engagement and interaction. Instructional designer and Web-based course instructors spend infinite amounts of time and effort to create on-line environments where learners feel they belong to the virtual classroom, but what of the student who cannot find the "classroom?" Designers and instructors ensure that learners can interact freely with the instructor and their classmates. They give on-line access to reading resources, support materials, and virtual libraries. Designers can even create clever Java applets, animating 3D representations as learning objects, but may fail to consider whether the learner's computer will support these electronic "aids to learning." As instructional designers, we invest ourselves in carefully posing electronic interactions and open-ended questions designed to stimulate higher-order thinking, but may not consider the variety of learner computer systems and learners' skills in using them. We measure cognitive skills and learning styles, but make assumptions about computing skills.

Web designers are able to show complicated technical imagery, but may ignore the fact that it takes 20 minutes to download that applet over a dial-in modem or that the network server may be unavailable at some time. Often, web designers and developers simply assume that the statement specifying the minimum system required to run the courseware is all that is needed and that the learner will easily and quickly comply. This author experienced one example of this phenomenon while working on an extensive project funded by a grant from the Alfred P. Sloan Foundation.
The Project

The project was a graduate level program in an engineering discipline intended to satisfy the requirements for professional certification. The program consisted of a series of five courses, four of which was composed of five units, each unit containing five lessons. The engineering content of the project was complex in itself, but the number of components required to deliver the instruction increased the difficulty of the project. The task of creating equations as gif format graphics files alone accounted for more than 1200 files for one unit. For each unit there was a printed study guide, simulated laboratory exercises, a collaborative activity based on asynchronous conferencing software, and the HTML-based instructional modules to coordinate and deliver the content instruction. A CD contained the HTML instructional units, the simulated laboratory and the additional browsers and clients…everything the learners would need.

The initial design and development team for the project prototype included five faculty members, a project manager, an instructional publications designer, two instructional designers (one for the asynchronous, collaborative learning component, and one for the WBT component). After the project was underway, a designer for the laboratory simulations was added. One would imagine that a team of this size could anticipate every learner problem and solve it in advance.

While pilot testing the project, those providing tech support were amazed at how much time was required by the learners to just "get up and running." While the step-by-step instructions were all in the printed manual, the sample audience was generally unable to: (a) determine whether their computer system complied with the minimum requirements listed in the manual, (b) identify what operating system they were using and therefore could not determine which version of the browser to select, (c) determine whether or not their computer had a sound card and, if so, how to adjust the volume, (d) copy files from the CD to the hard drive, or (e) reset the screen resolution. Telephone help was required to accomplish these tasks that many of us take for granted. In one instance, a custom install CD had to be created for one particularly inexperienced learner.

With all this, it is not hard to understand how some were close to withdrawing from the pilot test. So, whose fault is it? The technically informed might look down their noses and berate the
learner for being so ill-informed, but are designers not responsible for identifying the learner characteristics before designing the instruction?

Beer (2000) says clearly that computer instruction may not be appropriate for novice computer users who have little or no experience using the World Wide Web (WWW) and points out that not all learners who would benefit from the training have access to computers in their workplace. Many office networks have firewalls that prevent two-way computer interactions. Therefore, we must first determine what is the appropriate audience for not only the subject matter, but also the level of technological user expertise required, or conversely we must select the technology expertise appropriate for our target audience.

What to Do Then?

It appears that the designer has a couple of choices. One choice is to simply specify a minimum system for the WBT/TBT. By specifying the minimum system requirements to access the instructional/training materials, the designer’s job is made much easier. After deciding to require a minimum system for the training, one just defines the "minimum system." No more wondering about whether this will work or not on the user’s end.

However, by taking this approach, designers create a class separation of technology "haves and have-nots" or "knows and know-nots." The opening screen or promotional materials may state in plain terms that there is a minimum system for which this instruction or training has been designed and if you, the learner, do not have at least that, you cannot learn this content. You may be discriminated against for simply using a Macintosh instead of a PC.

There are always those determined learners who will try to proceed anyway and may possibly learn in spite of it all. One inexperienced student in a WBT pilot test clicked on all the icons on the course CD which loaded three different versions of the software, mixing 16-bit and 32-bit components, none of which ran properly. If the needs assessment or marketing research indicates many of the members of your target audience fall into the beginner or novice category, then a non-
technology delivery might be more appropriate. Remember - the instruction drives the technology, not vice versa.

After establishing an appropriate minimum system definition, one must determine how the instruction will be best delivered using Web technologies. Here, standard principles of good design apply:

- Keep file sizes small for short download times.
- Use technology appropriately (just because you can, does not mean you should).
- Design the instruction first, not the web pages.

**File Size**

File sizes should be kept as small as possible. Learners have little patience with downloading files and the faster, the better. If a file must be large, make sure there is something to inform the user what is happening and keep their attention during the download. Inexperienced users may think something is wrong and click about on the page, starting and stopping Web access and exacerbating the situation. Use compressed files when possible, but if a file will take a while to decompress, let the learner know and keep something happening on the screen to let the learner know the status of the file download or decompression. The author has seen a variety of entertaining “busy signals” that display while files are downloading.

**Gratuitous Technology**

Yes, there are so many really exciting things the Web can do and one wants to use them all. There is cgi, Java, Javascript, streaming audio and video, XML, animations, and so much more. How can one cram it all into one really cool webpage of instruction? Before beginning a "cram session," think about the learner characteristics and the minimum system definition. Will the learner have the right browser? Are the necessary plug-ins installed and "helper apps" available? Is Java enabled? Is there any way to get answers to these questions without alienating the learner from the start?

Good instructional theory tells us to keep the learner engaged and motivated and that a variety of stimuli can achieve this goal. On-line, however, these efforts to engage may actually backfire,
slowing computer performance to a crawl. The instruction not only loses the learner’s attention, but causes him to lose patience and perhaps temper. Learners will leave a computer environment faster than a face-to-face classroom. They get up and walk away frustrated, tainting the experience with the frustration. In presenting a seminar on "TechnoStress," the author heard many stories of computer learners completely abandoning an on-line tutorial due to frustration from one source or another.

More than once, this author has encountered a situation where a "simple" Java applet or a specially designed Web activity resulted in a confused learner. The learner sat, staring at a computer screen where nothing happened, because there was not enough memory available to run the applet. One answer is to include an error message if the applet fails to load. Loading interlaced graphics may help as well. Keeping the learner informed mitigates anxiety and frustration.

If the designer determines an added media component such as audio or video is essential for the learning experience, she must provide an alternate way for learners without these capabilities or skill levels to access that activity. If your site uses frames, include a non-frames version and have the script determine which is appropriate automatically. Why ask beginners to determine whether their browser supports frames or not? They have enough to think about already. Just because more people have computers, it does not mean more have mastered them.

**Think Instruction First**

Choosing buttons and arrows first is analogous to choosing curtains and wallpaper before the house is started. It is more important to know how many walls and windows there will be than what color the curtains are. The colors and schemes may sound great in theory, but in the house, they may not really fit. Beer (2000) reminds us that the "learning flow, not the page flow, should be what determines your presentation. You need to figure out what content and tasks are needed, and in what order" (p.94). Then and only then can one decide what buttons or arrows or links or colors are appropriate.

WBT requires a more thorough assessment of learner characteristics. In addition to the commonly addressed characteristics such as reading level, cultural background, age, gender, and
content experience, one must assess the level of computer competency as well. Technology is supposed to be transparent in instruction. One way to create transparent WBT is to use methods that determine the user skill level or needs of the learner through a series of questions (wizards) or to include coding that interrogates the machine directly to determine the appropriate settings and displays.

In truly identifying the learner characteristics, one needs to assess both the learners' computer skill levels as well as their computers' capabilities. What speed is the microprocessor? Does the operating system support your technology? Does the browser support your webpages? Is the browser Java enabled? Will it handle JavaScript? Frames? How does it handle graphics files and video? Is the proper version of Quicktime loaded? Are the helper apps and plug-ins in place? What screen resolution is needed? How many colors does it support? What speed modem are they using?

The author requires her WBT classes to go through several WBT tutorials at the beginning of the class, so they experience WBT from the learner’s viewpoint. They record their successes and frustrations and discuss them in class. They are told to make sure those things do not occur in the WBT projects they create. Their frustrations make them more sensitive to the frustrations of others and more determined to avoid those pitfalls.

**Conclusions**

There can be many pinnacles in Web-based (WBT) or Technology-based (TBT) Training, but there can be devastating pitfalls as well. Designers actually create two environments every time they design and develop a WBT module—a learning environment and a technology environment. When the learner sees the technology instead of the “teachnology” of the learning environment, the learning suffers. A learner’s first experience with WBT determines the degree of willingness to continue learning in this mode. As designers, we must attend to the myriad technical variables that impact upon the learning environment and do whatever is necessary to assure the transparency of the technology, both in the design of the user interface as well as in the behind-the-screen operation of the WBT.
References


