DOCUMENT RESUME

ED 392 393 IR 017 628

AUTHOR Misanchuk, Earl R.; Schwier, Richard A.
TITLE Benefits and Pitfalls of Using HTML as a CD-ROM Development Tool.
PUB DATE Feb 96
NOTE 20p.; Paper presented at the Annual Meeting of the Association for Educational Communications and Technology (Indianapolis, IN, February 12-16, 1996).
PUB TYPE Guides - Non-Classroom Use (055) -- Speeches/Conference Papers (150)

EDRS PRICE MF01/PC01 Plus Postage.
DESCRIPTORS Authoring Aids (Programming); Computer Graphics; *Design Preferences; Foreign Countries; Hypermedia; Instructional Design; Instructional Materials; *Multimedia Materials; Nonprint Media; *Optical Data Disks; Programming Languages; Screen Design (Computers); Visual Aids
IDENTIFIERS *HTML; HyperCard; Limits to Growth; Visual Displays

ABSTRACT

The hypertext markup language (HTML) used to develop pages for the world wide web also has potential for use in creating some types of multimedia instruction destined for CD-ROMs. After providing a brief overview of HTML, this document presents pros and cons relevant to CD-ROM production. HTML can offer compatibility to both Windows and Macintosh platforms, hypertext capabilities, a high degree of learner control over display characteristics, and a relatively small number of commands to master. On the other hand, HTML is not an authoring tool with the versatility of HyperCard and other similar aids; it is inflexible enough to limit control over interactivity and layout, and complicated enough to cause confusion at many stages in the process. Because HTML is a still-evolving standard, editing and conversion tools, reliable information, and even good advice are often hard to come by. Hardware and software requirements can be substantial, the UNIX-based environment is not always user-friendly, and special characters may not be readily incorporated into an HTML document. These drawbacks specific to HTML itself may be lessened with time, but even if they are surmounted, the designer must still be on guard against making poor design choices, such as making graphics too ambitious and fancy at the expense of informational content, using too many textured backgrounds, or choosing color combinations that hamper readability. (BEW)

Benefits and Pitfalls of Using HTML as a CD-ROM Development Tool

Earl R. Misanchuk
Extension Division
University of Saskatchewan
Saskatoon, SK S7N 5C8
Canada

(306) 966-5555
Fax: (306) 966-5557
Earl.Misanchuk@USask.Ca

Richard A. Schwier
Curriculum Studies
College of Education
University of Saskatchewan
Saskatoon, SK S7N 0X1
Canada
(306) 966- 7641
Fax: (306) 966-7658
Richard.Schwier@USask.Ca

Paper presented at the Annual Convention of the Association for Educational Communications and Technology, Indianapolis, IN, February 12-16, 1996

The rates of dissemination and adoption of the World Wide Web (WWW) have probably been greater than those of any other technological phenomenon in history. It is unusual to get through a day without hearing or seeing some reference to the WWW. Naturally, educators seek to exploit the capabilities of the web for their purposes, and many individuals, as well as schools and other organizations, are actively involved in developing pages accessible via that application of the Internet known as the WWW.

It may not immediately be obvious to multimedia designers that the HyperText Markup Language (HTML) used to develop pages for the WWW also has potential for being used to create some types of multimedia instruction destined for CD-ROMs. While there are some substantial advantages to using HTML rather than tools such as HyperCard® or ToolBook™ for CD-ROM production, there are also some limitations and potential pitfalls of which instructional developers must be aware. This paper identifies and elaborates the benefits, the limitations, and the pitfalls to avoid.

Overview of HTML

It is not our intent to describe HTML completely here, or even to describe most of it. There are extensive references available that do that, both on the WWW (e.g., see
http://www.ncsa.uiuc.edu/General/Internet/WWW/HTMLPrimer.html#A1.1;
http://oneworld.wa.com/htmldev/devpage/dev-page.html;
http://www.w3.org/hypertext/WWW/MarkUp/MarkUp.html)
and in print (e.g., Graham, 1995; Savola, 1995). Rather, we intend to describe just enough about HTML that a complete novice can follow the discussion below.

HTML consists of a collection of "tags" which enclose text to achieve certain effects. For example, to cause the word "jungle" to be rendered in boldface on the learner's screen, the HTML specification would be <B>jungle</B> . Similarly, to cause the words "Time is Money" to appear centered on the learner's screen, the specification would be <CENTER>Time is Money</CENTER>. Usually, tags can be combined (e.g., <B><CENTER>Time is Money</CENTER></B> would cause a bolded, centered "Time is Money"), although there are strict rules governing what is and what is not acceptable syntax.

Hypertext links are created between words or graphics with similarly arcane syntax (e.g., <A HREF="http://www.extension.usask.ca/Graphix/Earl's Face">See my face</A> translated into human language says "When the learner clicks on the phrase "See my face", connect to the WWW server whose name is http://www.extension.usask.ca, locate the folder or subdirectory entitled "Graphix", within which you will find the file named "Earl's Face", and display the latter on the screen").

Needless to say, HTML can be somewhat difficult to read in its raw form. Figure 1 shows the HTML code for the contents of Figure 2. As HTML and the WWW become more mainstream, we can expect a great deal of HTML to be (thankfully!) hidden from users, just as the evolution of word processors has hidden the tags that were used with early versions of that type of program.

- -

Insert Figures 1 & 2 about here

- -

The WWW, like the rest of the Internet, works on the basis that the learner operates a computer (known as the client) which connects to another computer containing the desired information (known as the server). The client software, known as a browser (Netscape, Mosaic, and lynx are some examples), works in conjunction with the server software, causing the desired information to be displayed on the client machine.

In "normal" use, HTML is used to create files that fit together seamlessly, no matter whether they reside on the same server or on another server. That is, the learner viewing the screen cannot ordinarily tell whether the next file to which she links is on the same server as the file she just saw, or on another server half the world away (hence the name World Wide Web).

However, it is possible to use HTML in another way—one that doesn't involve the usual client/server relationship operating via a connection to the Internet. In this application, the learner's computer effectively acts as both the client and the server, and permits the learner to access files stored either on her computer's hard drive or another device connected to her computer, such as a CD-ROM. The same kind of browser program as used on the client machine in the "normal" case is required, but no server software is needed. Thus the browser program can do double duty—as a means of accessing the WWW and as a means of accessing CD-ROM-based material.

Benefits of Using HTML

Cross-Platform Compatibility

Perhaps the greatest benefit of using HTML as the basis for CR-ROMs is that the same set of text, graphics, and sound files can serve both Macintosh and Windows platforms; only the user-provided browser program needs to differ. Hence no file-conversion or -duplication is necessary. This can represent a substantial saving of time and effort to the would-be CD-ROM producer.

It is worth noting that some modification of file naming conventions may be necessary if cross-platform compatibility is your goal. HTML files by convention must bear the suffix ***.html***, and video files in JPEG format must carry the suffix ***.jpeg***. PC-based web servers can only cope with three-character extensions to file names, so a shortened form of extension must be used: ***.htm*** and ***.jpg*** or ***.jpe***, respectively. Macintosh servers, and Macintosh and UNIX browsers that we have tried, seem able to cope with the three-character extensions, so a good rule of thumb is to use the short form of extensions at all times.

Hypertext Capabilities

As already mentioned, HTML was specifically designed to permit hypertext links similar to those in HyperCard or SuperCard on Macintosh and ToolBook on Windows. Links can be created between text displays or between pictorial or video displays.

Text, Graphics, Audio, and Video Can be Incorporated

HTML documents can incorporate text, graphics, audio, and video. The basic browser programs can only decipher and render text and a restricted range of graphics formats; "helper" application programs that work with the browser are needed to articulate audio, video, and some types of graphics. (Some examples of such helper applications are JPEGView, SoundMachine, and Sparkle.) Fortunately these helper applications are readily available, usually at little or no cost to educational institutions (as are the browsers themselves).

High Degree of Learner Control over Display Characteristics

Learner control over various aspects of computer-based presentations has been shown by research to be beneficial, and HTML browsers such as Netscape give the learner much control over what she sees and hears. Here are just a few of the features over which user control is provided in the most recent version of Netscape:

- font in which basic text material is displayed
- size of font
- color of font
- color of background
- color of links and/or underlining of links
- differentiation of "followed" links from "fresh" links
- time to expiration of "followed" links
- location and size of viewing window (with scrolling available to view images larger than the current window)
- graphics reception (browser program preferences can be set to either avoid downloading graphic images automatically—which can save time if you don't care to see the images—or to download them automatically)
- display images while loading or after loading (since graphic images are interlaced, the former permits a sort of preview of the graphic as it is being loaded, permitting the learner to cancel further loading of the page if the graphic is of little interest rather than waiting until the entire graphic has loaded; displaying graphics after loading provides all the text material first, then the graphics)

Designers can exercise a limited degree of control over the way text and visuals are arranged on the screen. They can, of course, determine what text is displayed and the structure of the text in terms of paragraphs (using either of two commands, <P> or <BR>). They can also insert headings of different size and location (left, center, right), and cause text to be displayed in list or "point" form (with or without numbers or bullets in several shapes). Designers can also cause portions of the text to appear in italics or boldface. Other character highlighting elements exist and still others are proposed for future versions of HTML, but are not yet part of the standard. Not all browsers support all the existing styles, however, so they should be used cautiously. (One highlighting element we hope you will avoid like the plague is <BLINK> </BLINK>!) There is only a limited amount of designer control possible over margins, via the <PRE> tag. Placement of graphics can be controlled within limits (i.e., they can be placed at the left, at the center, or at the right of the page), and text can be caused to align with them and to flow around them in a few different ways. Control over vertical spacing is very limited (which point will be discussed later in this paper). Horizontal rules (lines) of varying widths and thicknesses can also be placed among the text by designers.

However, given the amount of learner control over what is displayed (described above) and the limitations of HTML itself (elaborated below), instructional designers may feel constrained in the degree of control they exercise over layout. Obviously, control over the text and headings, whether or not graphics or sound or video are made available and whether they are mandatory or optional, what hypertext links exist, and other such decisions are within the scope of the designer of an HTML document. However, more subtle decisions such as what fonts and font size are used are generally beyond the control of the materials designer, since the user has control over them. Of course, HTML can be "tricked": if you *really* want to dictate exactly what font the learner sees, you can create a graphic image that contains the text you want, the way you want it. Doing so, however, not only subverts the original intention of HTML (to give the user more control) but also imposes the penalty that graphics files are much larger and therefore slower to load into the learner's machine than text files.

Relatively Few "Commands" to Learn

There is a relatively small number of "commands" (more correctly, elements and markup tags) available in HTML, making it easier to learn than a programming language that has many commands. However, as noted below, saying that HTML is *easier* to learn than some others is not the same as saying HTML is necessarily *easy* to learn.

We have seen that HTML offers a number of potential advantages to designers of instruction, and that it can be used for locally-stored information as well as for information distributed on the WWW. However, HTML is not a panacea, and there are some potential problems of which a would-be designer should be aware. It is difficult to distinguish between limitations of HTML per se and some of the problems currently encountered when trying to implement instructional materials with HTML. Nevertheless, we have attempted below to make that differentiation by labeling the former category of problems "limitations" and the latter category "pitfalls".

HTML is not an authoring tool, although it can mimic some aspects of authoring. In attempting to use it for designing instructional materials, we are asking it to do a job for which it was not designed. Thus it would be unfair to suggest that HTML is somehow lacking; it does what it was intended to do quite well. It only falls short when it is applied to something it wasn't primarily intended to be able to do (just as a word processor can emulate certain basic functions of a database program, but can't do it as elegantly or as completely).

Limitations of HTML for Designing Instructional Materials

Some of the advantages of HTML noted in the last section are double-edged swords: they can equally be viewed as disadvantages. Among these are that:

- relatively few "commands" are available in HTML (while advantageous to ease of learning, the fact that relatively few "commands" are available means that one cannot exercise the same degree of control over interactivity and layout as if there were more available);

- some parts of the ultimate display format are under the control of the user, not the designer (which limits the control the designer can exercise over what the learner will experience); and
- helper applications are used to support certain file formats (if the learner doesn't have a viable copy of the necessary helper application, the full experience will not be available).

HTML has Limits to its Interactivity

HTML is not set up to be highly interactive; it is primarily an information display technology rather than an instructional technology. As will be explained below, however, HTML is an evolving standard, and already we are seeing extensions and adaptations that hold interest for designers of instructional materials (e.g., HotJava). Nevertheless, HTML is not likely to ever become as versatile an authoring tool as Authorware, for example, or even HyperCard or ToolBook. Simply put, there are lots of things an instructional designer might wish to do that HTML cannot handle, such as receive learner input in a wide variety of ways (e.g., dragging an object) or evaluate and respond differentially to learner input (e.g., provide feedback on the correctness of answers).

HTML Limits Control over Layout

HTML, by virtue of giving much control of the display to the learner, imposes some limitations on layout. Designers are accustomed to arranging elements such as titles, text, and graphics on a page in a certain way; indeed, such layout decisions can contribute to effective learning. However, HTML removes a good deal of that kind of control from the designer—sometimes to the annoyance of the designer. For example, HTML imposes quite rigid strictures on both horizontal and vertical space. While five consecutive strikes of the space bar will produce a corresponding horizontal space on the printed page, on HTML documents five consecutive spaces will be rendered as a single space. Similarly, striking the carriage return key ten times will cause a vertical space of significant size in printed documents; in HTML ten equivalents of the carriage return is not only *not* rendered as a sizable vertical space, it is rendered as a single horizontal space. (One workaround to this problem is to intersperse paragraph markers (<P>) with periods, thus: <P>.<P>.<P>.<P>, which will give a vertical space equivalent to four carriage returns. Of course, the periods will also appear on the screen, but since they are so small, they will be hardly noticeable.)

Tabs have no effect on HTML documents (more correctly, one or more tabs are rendered as a single horizontal space, one character wide).

The various levels of headings in HTML are neither designer-controllable nor user-controllable. That is to say, one cannot increase the size of the font used for the most major heading, except by the user increasing the size of the body text (which then adjusts the size of the headings correspondingly). A designer can, of course, "cheat" and choose a lower-level heading, thereby reducing the font size of the heading relative to the body text, but HTML has a built-in control over increasing font size in headings.

If a large figure, to which the text makes reference, is included in a document, the learner may have to scroll up and down between text and graphic.

Helper Applications may be Required

Another limitation of using HTML for CD-ROM-based multimedia instruction is that users must have appropriate helper applications to work with the web browser, such as Sparkle to play MPEG movies or SoundBlaster to play sound files. As noted above, helper applications are quite widely available and, for educational institutions at least, are free or very low-cost. Still, the fact that HTML-based instructional materials on CD-ROM might require helper applications can be a limitation and there is no guarantee they will stay inexpensive or free. The very fact that the helper applications are "add-ons" to the browser program could be problematic: Learners might be confused by the relatively terse and generally uninformative messages that the browser produces when it looks for a helper application and cannot find it.

Producing Graphics for HTML documents can be Complicated

Another current limitation of HTML centers around available methods of producing graphics. Because HTML recognizes only certain file types, getting from your favorite graphics program to an HTML-acceptable format can be less than straightforward. So-called "in-line" graphics (those that do not require helper applications) can be of only three file types, of which GIF is probably the best known. Not all graphics programs will permit saving files in GIF format, hence conversion programs may be necessary. (This should change over time as newer versions of graphics software begin to support that format.) Higher resolution images, and those with greater color depth (i.e., more "shades" of color) require a different format, JPEG, which in turn demands a helper application to render the image on the screen. Once again, a conversion program may be required in order to save a file in JPEG format.

Hypertext can be Confusing

Getting lost in hypertext is a well-known phenomenon, especially among learners new to the format. HTML, designed specifically to provided hypertext capability, is no exception. Extra care must be taken to provide learners with understandable routing choices and mapping capabilities.

Pitfalls to Using HTML

HTML is Not an Authoring Language

As noted earlier, HTML is not an authoring language and anyone approaching it with the misapprehension that it is, will likely be dismayed by the relative lack of versatility afforded by HTML as compared to HyperCard, ToolBook, and similar development tools. HTML is reasonably good at dispensing and displaying information but is hampered by a limited range of interactivity. While it is true that information can be collected from users through the use of forms, dealing with that information once obtained is an idiosyncratic activity. If and when generalized tools become available for manipulating form-collected data, this pitfall may be ameliorated, but until then designers have to recognize that they are unlikely to be able to go far beyond "page-turning" designs (albeit with hypertext capabilities) with HTML.

HT'ML is an Evolving Standard

A significant pitfall of HTML is that it is an evolving standard—a "moving target" that makes it difficult for a designer to know what features it is safe to include in an instructional package. At this writing, HTML 2.0 has been specified. That is, there is an "official" specification of what HTML should be capable of doing and the manner in which it should do it. Given the nature of software development, it should come as no surprise that version 3.0 is under very active development. Browser programs generally support HTML 2.0 (the "official" version) but sometimes have partial implementations of some characteristics of version 3.0, or at least what is expected to become part of the "official" version 3.0.

The fact that some producers of browsers have taken over the leading edge of development on certain fronts exacerbates the problem. Can a designer safely include tables in a package? Will the "official" standard eventually catch up and include tables? If so, will the standard match the existing reality? If not, what will happen to learners who try to use browsers other than the one with which the materials were developed?

Good Editing and Conversion Tools are Scarce

Although the situation has improved over the past year, reliable and seamless editing tools for HTML are still forthcoming. A spate of shareware editors have been available for a year or so, and some have become sufficiently stable and popular that they have evolved into commercial products. Virtually all of these editors still employ tags as visible entities (i.e., the tags still appear on the screen, adding visual clutter to the text to be displayed). Some mainstream word processors have recently included or are about to include HTML as an optional output format, hopefully thereby removing the necessity of writing and reading tag information in its raw state. It remains to be seen whether the HTML document designer can be entirely sheltered from tags, however, especially when editing existing documents.

Conversion of existing word processing documents to HTML format is a similar challenge. Although several filters and conversion utilities exist for this purpose, we have yet to find one that applies universally, reliably, seamlessly, and accurately.

File Size Drastically Affects Display Rate

Strictly speaking, this is not a limitation of HTML per se; still, it is something any would-be designer must keep in mind: File size markedly affects speed of access. Since movies, graphics, and often sound files tend to be large, and they have to be first loaded, then played, the learner may find herself doing little but waiting for the computer to catch up. Seamless designs of instruction are difficult to achieve. This problem occurs regardless of whether the HTML is resident on the WWW (where in addition to file size, maximum speed permitted by the network hardware and busy-ness of the WWW also affect delay time) or on CD-ROM (where computer processor speed, throughput speed, and speed of the CD-ROM drive itself can have effects).

Good Information on HTML can be Difficult to Find

Just finding information on HTML has been a pitfall in its own right. Since HTML is an evolving standard, merely keeping current with what features are available to the designer is a problem. The information exists; it's out there on the WWW; but how do you go about locating it? There are several WWW sites that purport to tell the novice everything that needs to be known in order to start developing HTML pages, but all too often they are merely re-statements of the official syntax specifications for HTML or collections of pointers to other sites. Perusal of them can leave all but the most computer literate confused and uncertain about how to implement certain HTML features. This situation is improving, ironically due to the publication of good books dealing with the topic (e.g., Graham, 1995; Savola, 1995).

UNIX is Not User-Friendly

HTML and the WWW started life on UNIX machines, and the arcane syntax of that operating system can be a real barrier to web page designers. Reading URLs (Universal Resource Locators, which are really just extended file names that describe the full path to the location of the file) employed by the WWW is a necessary skill, but a user-unfriendly one, especially to people accustomed to thinking in terms of Macintosh or Windows folders. Writing the UNIX specification for linking to a file in another folder can be daunting and confusing. Linking a file within a folder to another file within another, hierarchically parallel, folder can be even more so. Although WWW server software has now been developed for Macintoshes and PCs as well, the concept of the URL must remain (to retain interoperability with the UNIX forebears). Navigating among subdirectories as necessary to specify URLs can be challenging for the beginner. The "School of Hard Knocks" has taught us that understanding URL structures and functions is essential at the outset of a project. Take the time to learn the different uses of partial or relative URLs and full HTTP URLs.

Be aware that while some characters (e.g., a space) may be allowed in filenames on Macintosh servers and browsers, they are disallowed on other platforms. Good HTML editor applications will usually take care of the problem by inserting HTML special-character equivalents to disallowed characters, but confusion may reign if awareness of this limitation is lacking. For example, a space is encoded in a URL as %20, so that a Macintosh filename like Design Ideas becomes Design%20Ideas. If someone is unaware of disallowed characters, the %20 may be mistakenly adjudged an error, and edited out. Similarly, be aware that URLs are very case-sensitive: capital and lower-case letters are not interchangeable.

Multiple Development Platforms May Cause Confusion

As noted above, the UNIX-based syntax of URLs can be confusing to neophytes. In order to circumvent this problem and make life easier for the designer, the developers of some HTML editor applications (at least on the Macintosh) implemented "point-and-click" methods of identifying the file to be linked to. That is, the editor application simply requires the designer to locate, through standard folder/file navigation procedures, the file to which the link is to be established, and then "captures" the file structure path taken to that destination file and records it. Unfortunately, if the development is being done on a computer different from the one on which the HTML files will eventually reside, those pathnames will be invalid. Changing machines may necessitate manual changes to the URLs, and automaticity would be lost.

Hardware and Software Requirements can be Substantial

Taking an idea from a word processor, moving it into an HTML editor, creating accompanying graphics, then converting all to appropriate file formats for WWW use, and finally linking the graphics files to the text files requires both a number of different application programs (hence much RAM) and the ability to move quickly and seamlessly among them. While HTML development can be done successfully on lower-end machines, it is difficult to do so efficiently.

Practical Advice can be Hard to Come By

There are a handful of file formats that will work on WWW, but precious little advice on where and how one of them would be more useful than another. For example, both SND and AU file formats are acceptable for sound files. Which one should a developer use? Is there a penalty for choosing the wrong one? The same kind of decision must be made between MPG and MOV files for movies, and for GIF and JPG files for graphics. A companion paper (Schwier & Misanchuk, 1996) begins to make some inroads to answering questions like these, but much work remains to be done.

Some Features cannot be Implemented without Server Software

In applications such as HyperCard and ToolBook, areas of the screen can be sensitized to a mouse click, so that a learner need only point at an object and click on it to cause something to happen. A similar capability, called image maps, exists in server-based HTML. However, we are unaware of any similar capability that will permit a single computer, being used in standalone mode (i.e., not connected to a separate HTML server) that will permit the use of image maps, which effectively rules out the use of image maps with CD-ROMs.

Special Characters

Special characters ("curly quotes", em and en dashes, ampersands, etc.) are not always transportable between word processors and HTML editors/readers. Although provision is made in HTML for such special characters, some browsers and/or HTML editors have apparently not given a high priority to implementing their interpretation flawlessly. Undoubtedly this problem will disappear with time, but for the moment, it might be wise to stick with the vary basic character set (ASCII) when creating pages.

Some General Pitfalls

Below are some pitfalls in designing screen-based displays that are not necessarily limited to HTML. We have, however, seen so many examples of those pitfalls on the WWW that we felt it worth identifying them here.

The Fancy-Graphics Pitfall

It is all too easy to fall into the trap of using gratuitous images because it is possible to do so and because "they look cool". When Macintosh first came out more than a decade ago, people were given—really for the first time—the opportunity to easily include as many different fonts on a page of text as they wanted (true, the IBM "golf-ball" typewriter did the same, but you had to work harder at it). Some went hog-wild with the new-found freedom. It took quite a while for the typical user to come to understand that more than a couple of different fonts per page is not only aesthetically lacking but quite possibly dysfunctional; some still haven't made that apprehension! So it is with including graphics, movies, and sound files within WWW pages and (by extension) on CD-ROM materials. There is a well-established body of literature that quite convincingly shows that pictorial material must be germane to the text surrounding it, and that the text must make reference to the pictorial material before it is useful (e.g., see Chapter 11 of Misanchuk, 1992). The notion of adding pictorial material to instructional materials for "interest" or "motivation" is often cited, but seldom supported by research. It is sobering to find, as some research has, that the inclusion of pictorial material actually acts as a distraction, and can thus be dysfunctional.

The "Non-Informative Information" Pitfall

"Surfing" the WWW for home pages of colleges and universities can be an enlightening experience. Notice how many of them have large, colorful pictures of some of their buildings as part of their home pages. Some of them can take as long as a couple of minutes to download, even on a relatively fast network connection.

Yet they offer no real information. Users who know the campus won't be impressed more than once or twice with the capability of seeing that building on their screens, and will likely become less tolerant of the extended download time if they access that page frequently to get to other information. Furthermore, users who don't know the campus will get relatively little useful information from the picture of the building.

Head-and-shoulder movies of college presidents reading a message of welcome (sometimes badly) also seem to be favorites. One wonders how often such "features" of home pages are actually used and appreciated by users. (Mission statements prominently displayed on institutions' home pages are another questionable device!)

Although we know from both the research available and from personal experience that research findings cannot always be generalized aci oss media, perhaps we should use as a starting point the large body of research that has been done on the utility of pictorial material for learning. At the risk of over-generalizing, it says (roughly) that pictures are useful only when they are salient to the content.

Textured Backgrounds

HTML permits the specification by the page creator of the color of the background on which text is displayed, thereby allowing the author to override user control of this element. Furthermore, it also permits text to be placed over a graphic; hence photographs and/or textured surfaces may be overlaid with text. Used judiciously, this feature can make for striking screen displays. However, in our experience, it is far more often abused to the extent that text superimposed on textures or on photographs can become all but illegible.

Poor Color Combinations of Text/Background

A number of legibility studies have been done regarding color combinations of text and background. Summaries of that research are available elsewhere (Misanchuk & Schwier, 1995a, b, c; Schwier & Misanchuk, 1995). Perhaps the most useful generalization to come out of those studies is that contrast between the text and the background is of paramount importance. Beyond that more or less "scientific" generalization lies the "artistic" one—the matter of taste. Certain color combinations look better to the eye than others, regardless of their legibility. Some generalizations in this regard are illustrated in Misanchuk & Schwier, 1995c.

Summary

While the HyperText Markup Language (HTML) can be used as a CD-ROM development and delivery tool with some significant advantages, it imposes certain constraints upon the designer of the CD-ROM. The most significant advantage is that HTML-based CD-ROMs can provide hypertext capabilities incorporating text, graphics, audio, and video on both Macintosh and Windows platforms with no file conversion or duplication necessary. Another advantage is that a high degree of learner control over presentation characteristics is possible. Furthermore, only a small number of HTML "commands" need to be learned in order to create simple documents.

However, there are some limitations to HTML as a CD-ROM development tool. Some of these limitations are inherent in HTML; others are pitfalls for the unwary that cannot be blamed on HTML, per se, but that can have negative consequences.

HTML was not designed specifically for producing CD-ROM based multimedia instruction, and therefore has some limitations when used that way: the kinds of interactivity possible are somewhat limited in comparison to software created with authoring programs; designers may feel a lack of control over the layout of content under HTML; helper applications to the basic browser program may be required to make HTML CD-ROMs fully functional; producing HTML-compliant graphics can involve some effort and resources; and hypertext itself can be confusing to learners who have had little experience.

Some of the pitfalls awaiting a would-be HTML developer relate to inherent characteristics of HTML and its origin; others are more temporal in nature—HTML is still a recent phenomenon and some of the pitfalls should disappear with time. These pitfalls include the fact that HTML was not specifically intended to be used as an authoring language; HTML is an evolving standard; good editing and file conversion tools are still relatively scarce; file size can drastically affect display rates; good information on HTML can be difficult to find; the UNIX-based origin of HTML creates a user-unfriendly aspect to using HTML; using multiple development platforms may cause confusion; hardware and software requirements for HTML production can be substantial; practical advice may be hard to come by; some features of HTML cannot be implemented in a standalone CD-ROM application; and special characters may not be readily incorporated into HTML documents.

Finally, some pitfalls we have observed that are not specific to HTML but seem to appear in that milieu all too often include the propensity to include too-fancy graphics; the use of "non-informative information"; the use of textured backgrounds; and the choice of poor text/background color combinations.

References

Graham, I. S. (1995). *HTML sourcebook: A complete guide to HTML.* New York: John Wiley & Sons.

Misanchuk, E. R., & Schwier, R. A. (1995a). *Appropriate uses of color in multimedia screen design: An illustrated guide to research results.* HyperCard™ stack presented at the Annual Conference of the Association for Educational Communications and Technology, Anaheim, CA, February 8-12, 1995. (Available from http://www.extension.usask.ca/Papers/Misanchuk/AECT95/Stack.sea.Hqx)

Misanchuk, E. R., & Schwier, R. A. (1995b). *The art and science of color in multimedia screen design, part II: Science, myth, and legend.* Paper presented at the Annual Conference of the Association for Educational Communications and Technology, Anaheim, CA, February 8-12, 1995. (ERIC Document Reproduction Service No. ED 381 144) Available for downloading as a Macintosh Word 5.1 file from http://www.extension.usask.ca/Papers/Misanchuk/AECT95/MS95.sea.Hqx

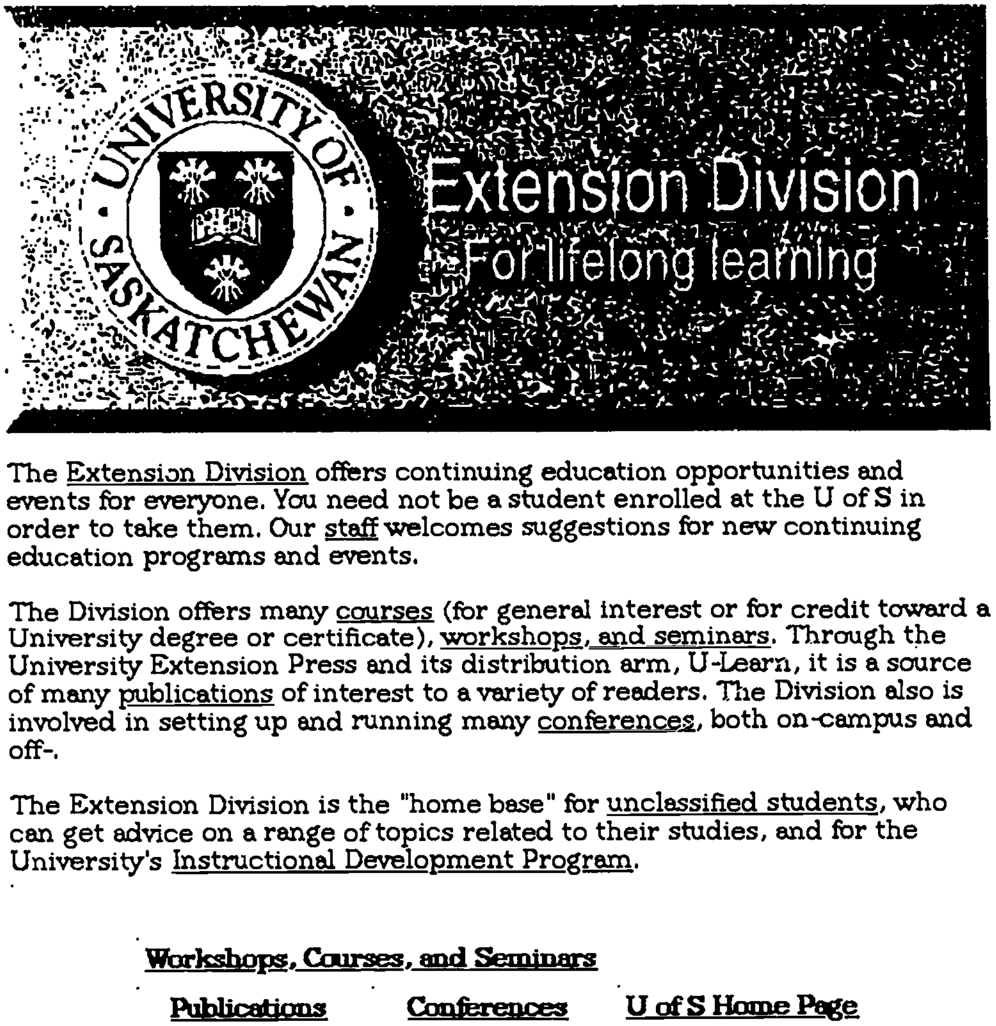
Misanchuk, E. R., & Schwier, R. A. (1995c). *The mythology of colour in multimedia screen design: Art, science, and connoisseurship. Canadian Journal of Educational Communication, 21*(4), 3-26.

Misanchuk, E. R. (1992). *Preparing instructional text: Document design using desktop publishing.* Englewood Cliffs, NJ: Educational Technology Publications.

Savola, T. (1995). *Using HTML (Special edition).* Indianapolis, IN: Que Corporation.

Schwier, R. A., & Misanchuk, E. R. (1995). *The art and science of color in multimedia screen design, part I: Art, opinion, and tradition.* Paper presented at the Annual Conference of the Association for Educational Communications and Technology, Anaheim, CA, February 8-12, 1995. (ERIC Document Reproduction Service No. ED 381 143) Available for downloading as a Macintosh Word 5.1 file from http://www.extension.usask.ca/Papers/Misanchuk/AECT95/SM95.sea.Hqx

Schwier, R. A., & Misanchuk, E. R. (1996). *Designing multimedia for the HyperText Markup Language.* Paper presented at the Annual Convention of the Association for Educational Communications and Technology, Indianapolis, IN, February 12-16, 1996. Available in electronic form from http://www.extension.usask.ca/Papers/Misanchuk/AECT96/Multimedia&HTML.html



The Extension Division offers continuing education opportunities and events for everyone. You need not be a student enrolled at the U of S in order to take them. Our staff welcomes suggestions for new continuing education programs and events.

The Division offers many courses (for general interest or for credit toward a University degree or certificate), workshops, and seminars. Through the University Extension Press and its distribution arm, U-Learn, it is a source of many publications of interest to a variety of readers. The Division also is involved in setting up and running many conferences, both on-campus and off-.

The Extension Division is the "home base" for unclassified students, who can get advice on a range of topics related to their studies, and for the University's Instructional Development Program.

Workshops, Courses, and Seminars

Publications **Conferences** **U of S Home Page**

Figure 1. Screen display of http://www.extension.usask.ca, for which the HTML code is shown in Figure 2.

```
<HTML>
<HEAD><TITLE>Continuing Education Opportunities
</TITLE></HEAD>
<BODY bgcolor="eeeeee" TEXT="#000000" LINK="#006600" VLINK="#8e2323" ALINK="#ffffff">
<BR>
<center><IMG  SRC="Graphics/Ext_Div_banner.jpg" ALT = "Extension Division Splash Screen"></center>


<p>The <A HREF="Ext_Div_desc.html">Extension Division</A> offers continuing education  opportunities  and events for everyone. You
need not be a student enrolled at the U of S in order to take them. Our <!– Link Tag -->
<A HREF="Ext_Div_desc.html#StaffLink0">staff</A> welcomes suggestions for new continuing education programs and events.
<p>
The Division offers many   <A HREF="Indexes/!WC&S.html">courses</A> (for general interest or for credit toward a University degree or
certificate), <A HREF="Indexes/!WC&S.html">workshops, and seminars</A>. Through  the University Extension Press and its distribution
arm, U-Leam, it is a source of many <A HREF="Indexes/Main_Pubs_Index.html">publications</A> of interest to a variety of readers. The
Division also is involved in setting up and running many <A HREF="Indexes/Conf_Index.html">conferences</A>, both on-campus and off-
.<p>
The Extension Division is the "home base" for <A HREF="Files/WCS/UnclassStudentAdv.html">unclassified students</a>, who can get
advice on a range of topics related to their studies, and for the University's <!-- Link Tag -->
<A HREF="Indexes/ID_Index.html">Ir.structional Development Program</A>.
<hr size=5><p>

<CENTER><TABLE CELLPADDING=3 BORDER=2>
<TR><TH ALIGN=CENTER VALIGN=MIDDLE COLSPAN=2>
<A HREF="Indexes/!WC&S.html">Workshops, Courses, and Seminars</A>
</TH><TH ALIGN=CENTER VALIGN=MIDDLE>

</TH></TR>
<TR><TH ALIGN=CENTER VALIGN=MIDDLE>
<A HREF="Indexes/Main_Pubs_Index.html">Publications</A>
</TH><TH ALIGN=CENTER VALIGN=MIDDLE>
<A HREF="Indexes/Conf_Index.html">Conferences</A>
</TH><TH ALIGN=CENTER VALIGN=MIDDLE>
<A HREF="http://www.usask.ca/">U of S Home Page</A>
</TH></TR>
</TABLE></CENTER>

<P><HR><HR>

</BODY> </HTML>
```

Figure 2. HTML code for screen displayed in Figure 1.

This paper is available in electronic form at
http://www.extension.usask.ca/Papers/Misanchuk/AECT96/Benefits&Pitfalls.html