Development of Concept-Based XHTML/CSS Pedagogy

for Non-Technical Learners

A Dissertation

Presented to the Faculty of the Curry School of Education

University of Virginia

In Partial Fulfillment

of the Requirements for the Degree

Doctor of Philosophy

By

Spring B. Brennan, B.A., M.A.

May 2014

Abstract

Spring B. Brennan, Ph.D.

Glen Bull, Ph.D., Chair

As the Internet continues to permeate daily life, the capability to create or manipulate Web pages is increasingly seen as a general computer skill needed by students from a variety of academic fields, not just those related to technology (Ariga & Watanabe, 2008). The disparate fields have incorporated Web design instruction, but the curricula have been vastly inconsistent, often depending too much on textbooks or software for their pedagogical framework. This Web design instruction has also been slow to integrate the instructional strategies endorsed by modern learning theories like constructivism, information processing and behaviorism, and as such, it has not met the needs of novices.

A curriculum unit was developed to teach Web design, XHTML and CSS to novices and non-technical learners. It strove to incorporate modern learning theory strategies, in particular the conclusions of the “new science of learning” (Bransford et al, 2000) which promote teaching for conceptual understanding and student-controlled learning. The project underwent an iterative piloting process, with revisions based on student outcomes and review by both subject matter experts and instructional design experts. This study then formally evaluated the curriculum unit and analyzed how best to apply the instructional strategies to Web design education. It also further documented the persistent misunderstandings that novices encounter while learning HTML, CSS and

Web design. Cascading Style Sheets (CSS) instruction in particular is an understudied topic, even though CSS is and will continue to be vital to the Web design process (Gillenwater, 2011).

A mixed-method research design employing classroom and video observations and document analysis of student Web pages was used to evaluate the curriculum unit. Six key findings on the unit's insufficiencies were identified from the data, all of which have implications for teaching Web design conceptually. There was considerable 'push back' from novices against the abstract nature of Web design topics, arguing a need for more procedural introductions to HTML and CSS before transitioning into conceptual learning strategies. The study also affirmed the importance of experience-building strategies for novices. For CSS instruction in particular, the use of demonstration/ modeling strategies, how non-visual HTML/CSS code translates to visual display, for example, was identified as especially important for building novices' conceptual understanding. The difficulties of teaching a techno-centric topic like Web design to learners with no prior experiences with computer languages or visual design are also discussed, as are the complexities of transitioning novices from procedural learning to conceptual learning.

Keywords: Web design, HTML, CSS, cascading style sheet, computer language, computer literacy, Internet literacy, visual literacy, instructional strategy, technology instruction, instructional design, self-instruction, self-directed learning, procedural learning, conceptual learning, constructivism, information processing, behaviorism

Dedication

For my mom, Laurie. For my good friend, Stephanie Fischetti.

Acknowledgements

The dissertation and XHTML/CSS curriculum unit benefited greatly from the feedback (and support) of my professors, reviewers and subject matter experts.

Special thanks to Dr. Glen Bull, whose Computer Courseware Tools assignments were the germ for this project and whose patience and support have been a continual boon, to Dr. Valerie Larsen for expert instructional design guidance and review, to Dr. Tonya Moon for direction on methodology design and learning theories, and to Dr. James Cohoon for computer science perspective.

Thanks also to Erika Powell and Sheila Warren, for serving as independent observers during data collection, to Dr. Daniel Tillman for serving (enduring) as instructor during the Fall 2009 piloting and to Dr. Ruth Ferree for coaching and mentoring during the dissertation writing process.

My appreciation also to the members of the External Review Panel, Dr. Katherine Neesen and John Rhea, for their time and expertise.

Finally, many, many thanks to the students whose assignments, questions and suggestions were used to revise and improve the curriculum unit and its implementation:

L. Apple, M. Bowyer, W. Bridges, E. Carson, J. Chao, J. Choi, J. Cohen, K. Couper, S. Demirkya, M. Dickerson, C. DuBos, J. Elliot, B. Harrison, W. Jones, W. Kjellstrom, H. Kwon, Y. Lee, V. Lewis, J. Ma, J. Madson, P. Malcolm, K. Mastrion, M. McDonald, P. McGuire, E. Pan, E. Powell, M. Remesz, H. Ritt, M. Semanik, K. Strazzeri, D. Tillman, S. Warren, K. Welsh, K. Wold, and Y. Yao.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

"The explosive growth of the Internet has made the knowledge and skills for creating Web pages into general subjects that all students should learn." (Ariga & Watanabe, 2008, p. 815)

The ever-advancing and time-compressed nature of Web technology leads to a lack of uniform definition for the term 'Web design' (Karper, 2004; Kotamraju, 1999). It is generally understood to mean the design, creation and posting of Web pages ("Web Design," Wikipedia, 2013), but the process can be as finite as formatting text and hyperlinks for push-button publishing (e.g., blogging) or as all-encompassing as the information design, graphic design, multimedia design, computer language markup, scripting and programming sequence necessary to produce a complex website or Web-based application (Burch, 2001; Hofstetter, 2006; Sklar, 2008; Teague, 2006; Wang & McKim, 2013). Regardless, as the Internet's presence in social life, work environment and school life remains continual, so does Web page creation activity. The Pew Internet and American Life Project reports that

> the proportion of adults who create or work on a Website (either a personal site, or someone else's) has remained consistent…Fourteen percent of online adults maintain a personal Web page (unchanged from the 14% who did so in December 2007), while 15% work on the Web pages of others (also unchanged from the 13% who did so in December 2007). (Lenhart, Purcell, Smith & Zickuhr, 2010, p. 25)

This positions Web designers or Web page creators in the minority, certainly. Overall, there are still far more Internet users 'consuming' Web pages than producing them (Hofstetter, 2006); "Content creation is largely the purview of experienced Internet users with high-speed broadband connections and ready access to the tools of content creation" (Karper, 2004, p. 61). But, these Pew statistics belie the increasing demand and expectations for Web page production; the expectation that anyone *can* and *should* create a Web page pervades, applying not just to technology-centric fields, but to people in every field. "Ordinary people do this," proclaims Erin Karper (2004) in her study of novice Web designers, and the expectation echoes, particularly in education: "The explosive growth of the Internet has made the knowledge and skills for creating Web pages into general subjects that all students should learn" (Ariga & Watanabe, 2008, p. 815).

Brief History of Web Design

The first Web page was posted on August 6, 1991 by Tim Berners-Lee, the physicist labeled as the inventor of the World Wide Web. His intent behind the project was to aid and inspire researchers by connecting them to information and resources in disparate locations (Berners-Lee & Fischetti, 2000). Titled simply "World Wide Web", the Web page consisted of text and hyperlinks presented in a linear fashion. (An archived version may be viewed at http://www.w3.org/History/19921103hypertext/hypertext/WWW/TheProject.html.) This first Web page was created with a single computer language, HTML, and its left-aligned text and layout offered little indication of the visual, functional and technological complexity which quickly dominated Web pages. As it stands now, "the Web is so far reaching in content and design that no collection of [Web] pages represents what is typical" (Sklar, 2008, p. 31). Web design progressed to this complexity by incorporating graphics, layout, multimedia, interactivity, and programming:

> Once upon a time creating Web pages was no more difficult than using a word processor. You learned a few HTML tags, created a few graphics, and presto: Web page. Now, with streaming video, JavaScript, ASP, JSP, PHP, Shockwave, Flash, and Java, the design of Web pages may seem overwhelming to anyone who doesn't want to become a computer programmer. (Teague, 2006, p. xi)

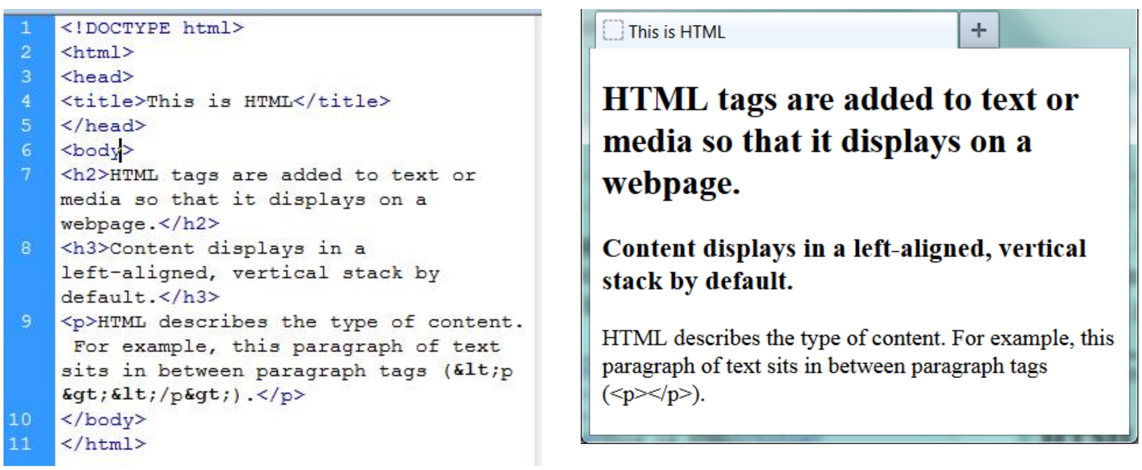

Figure 1. Example HTML (left) and how it displays in a browser (right).

An early contributor to Web page complexity were the attempts to make Web pages resemble print publications or documents with elements such as multi-column layouts. HTML, however, had not been created with graphic design in mind: "The HTML language, originally conceived to describe the structure of academic documents, was now being used as a page layout language—a usage for which it was entirely unsuited"

(Andrew & Yank, 2008, pp. 5-6). HTML layouts were initially achieved via HTML table markup, which proved problematic. HTML table elements were intended

> for the purpose of organizing tabular data into rows and columns. Web designers quickly realized they could use the table elements to build print-like design structures that allowed them to break away from the left-alignment constraints of basic HTML…This misuse of the table elements, although well-intentioned, has created problems with Web site accessibility and compatibility that are still influencing Web design today. (Sklar, 2008, p. 119)

Table-based Web design was then deprecated in favor of style sheets—additional computer languages like CSS—which added much more complexity to the process of creating Web pages, but allowed increased creative flexibility and control over presentation while avoiding HTML table issues (Andrew & Yank, 2008; Lie & Bos, 1999; Sklar, 2008, Wilcox, 2008). Cascading Style Sheets (CSS) were introduced in 1995 by Web browser developers Håkon Wium Lie and Bert Bos and were designed to facilitate the semantic separation of style from content (Lie & Bos, 1999; Powell, 2010).

> CSS…was designed to allow precise control—outside of HTML—of character spacing, text alignment, object position on the page, audio and speech output, font characteristics, etc. By separating style from markup, a web designer can simplify and make web contents more accessible at the same time. (Liu & Downing, 2010, p. 276)

(For example: CSS code creates the visual design, including the layout columns, while the HTML code places content like text and images on the Web page so the content will display in browsers.) CSS style sheets were quickly deemed the "saviors of responsible

Web design" (Collison, 2006, p. 3) and were embraced both because of what they made possible visually and because with each new version of the computer language, CSS's functionality and benefits grow:

> Style sheets aren't just useful for making attractive pages. By dividing structure and style, they can make documents simpler to create and easier to manipulate. CSS provides many valuable layout properties that provide a richer palette for design than presentation markup [HTML] ever could. (Powell, 2010, p. 519)

Figure 2. Example CSS and HTML (left) and how they display in a browser (right).

Using CSS3 (the current version), for example, "it's quite possible to make a graphically rich site that uses not a single image, drastically cutting the number of HTTP requests and increasing how fast your pages load" (Gillenwater, 2011, p. 17). CSS has even grown to encompass the creation of Web page animations, behaviors, visual transitions and transformations, as well as graphic design options that used to only be available in photography or drawing software—CSS3 can be used to add radiuses, shadows, gradients, and opacity changes to Web pages, for example (Gillenwater, 2011; Weyl, 2012).

Adoption of CSS by Web designers and developers has been slow and problematic, though (Andrew & Yank, 2008; Gordon, 2005; Hofstetter, 2006)—blame is most-often assigned to Web browsing software, which still fail to fully support CSS (Wilton-Jones, 2011):

> Browser support has been quite inconsistent, and significant bugs, particularly in older of versions of Internet Explorer, have made the use of CSS a lesson in frustration…even as CSS support has become more commonplace, significant issues remain. Browser bugs still exist, portions of the CSS specification remain unsupported, developer education and uptake is lagging, and proprietary extensions to style sheets are rapidly being introduced by browser vendors. (Powell, 2010, p. 430-431)

CSS's ease-of-use is also hindered because the language itself can be challenging to work with: "The problem with CSS is that CSS is too hard" (Andrew & Yank, 2008, p. 1). The logic of CSS can be difficult to understand, and

> as CSS was conceived in an age when the design of most web sites still looked quite plain, its creators couldn't anticipate the richness and intricacy of the designs that it would eventually be asked to describe...Clever designers figured out ways to make CSS do what they needed it to do, but these techniques were so convoluted that they quickly became difficult for the rest of us to master. (Andrew & Yank, 2008, pp. 1-2; see also Connolly, 2012)

CSS is conceptually difficult to master because it requires the designer to create visual design using non-visual computer language code (Powell, 2010)—its use would be analogous to requiring photographers to retouch or manipulate photos by typing computer languages instead of using photo-editing software. As with HTML before it, CSS ultimately wasn't created to address advancements in graphic design, especially layout: "Page layout with CSS [is] a black art that rarely [works] perfectly, predictably, or reliably, even for its most experienced practitioners" (Andrew & Yank, 2008, p. 13; see also Mills, 2013; Wilcox, 2008).

CSS is clearly still a work in progress—CSS3 is still not fully supported by browsers, but the first draft of CSS4 specifications was released by the W3C in 2011 (Gilbertson, 2011). But, CSS versions improve and become more powerful, browsers progress, and CSS is now vital to the Web design and development process (Connolly, 2012), working in conjunction with not just markup languages like HTML, but most other computer languages used on the Web or on mobile platforms (W3C.org, "HTML & CSS"): "Graduating from HTML-based formatting to CSS-based formatting is an important step for all hypertext authors" (Gordon, 2005, p. 64). Knowledge of CSS is also increasingly a required skill:

> CSS3 is not going away. This is how we're all going to be building sites in the future. Knowing CSS3 is an increasingly important and marketable career skill. Right now, it's something that sets you apart as a top-notch designer or developer. Sooner than later, it will be something that's expected of you. (Gillenwater, 2011, p. 19)

As the demand to create or manipulate Web pages increasingly becomes seen as a general computer literacy or as a “general [subject] that all students should learn” (Ariga & Watanabe, 2008, p. 815), then CSS will only grow as a required knowledge set for students.

Web Design in Education

Throughout its progression as a technological, visual, and communications process, Web design was embraced by the educational and academic culture, so that ultimately the creation and use of Web pages were “being studied critically and theoretically in disciplines as varied as communications, liberal arts, business, law, policy, and computer science” (Royal, 2005, p. 400). To this list, add journalism (Royal, 2005), rhetoric and composition (Dick, 2006; Karper, 2004; Turnley, 2005), education (Marx, 2003; Victor, 2002), information science, library science, linguistics, mathematics, psychology, and particularly, literacy (Hofstetter, 2006; Mackey & Ho, 2005).

There is little historical information on when Web design instruction was first added to higher education curricula, though. It likely was integrated after Tim Berners-Lee’s 1993 introduction of HTML or perhaps the 1993 introduction of the Mosaic web browser, “a graphical user interface that made the Web extremely easy to use” (Hofstetter, 2006, p. 13-14). Maddux, Liu, Cummings, and Smaby (2008) offer anecdotal evidence as part of their study:

> In 1993, because of increasing pressure on university teacher education faculty members to prepare preservice teachers to design, publish, and maintain their own educational Web pages, [we] began offering a course in Web design for teachers. (p. 4228)

Much of this early Web design was ad hoc, underdeveloped, and limited by Internet browsers still in embryonic form (Raggett, Lam, Alexander, & Kmiec, 1998).

By 1997 though, exploring Web design was recognizably added to existing curricula, again likely motivated by advances in Internet technology and HTML—this time it was the W3C-standardized HTML 3.0 ("HTML," Wikipedia, 2013). Self-instruction in HTML and other computer literacies also gained as educational practice, since increased Internet access and expectations motivated students to explore the new medium (Carter, 2006; Karper, 2004). For example, universities now offered space on Web servers to students and faculty for posting websites and "students [seemed] to be the most active group of home-page owners" (Döring, 2002). At this time, universities also added Web design software to their professional development, lifelong learning, or certification programs (Ariga & Watanabe, 2008), and began employing students to design and maintain university websites:

> Many college officials say they could never have put up their Web pages without help from their students, who know more about the Web than most administrators do, and are willing to spend the time it takes to create pages. (Fiore, 1997, p. A221)

As Internet software and browsers evolved, so did the curricula—WYSIWYG editors that visually-rendered Web computer languages (i.e., HTML and CSS) were added to or supplanted instruction on hand-coding:

> Web site design skills constantly incorporated new software essential to design…as the technologies of Web site design grow more complex, software to

> manage that complexity by automating aspects of Web site design emerged and became part of the repertoire of Web site design skill. (Kotamraju, 1999, p. 466)

Claims that “it is now possible for almost anyone to make sophisticated Web pages with word processor ease” (Descy, 1999, p. 5) or that “after just a few mouse clicks, scholars can create Web pages out of any research paper” (Hofstetter, 2006, p. 38) turned out to be overly-simplistic and overly-optimistic, but the reliance on Web design software and tools, and instruction on how to use them, flourished along with these attitudes. Finally, as Web page multimedia and interactivity demands increased, so Web design curricula swelled to include topics like creating Web databases, applications, video and audio (Gordon, 2005; Krunić, Ružić-Dimitrijević, Petrović, & Farkaš, 2006; Mackey & Ho, 2005; Whitehead, 2002).

What comprised a Web design curriculum was vastly inconsistent though—it depended on the academic discipline, whether the course focused on technology or information literacy, for example, or even the hardware and software available to students at the particular university (Stepp, Miller & Kirst, 2009; Turnley, 2005). Now into a third decade, this instructional inconsistency remains. University courses incorporating Web design proliferate, but there is seemingly little uniformity to how they are taught or their academic rigor. Many simply take the form of software training, treating Web editing software tools (e.g., Adobe Dreamweaver) as synonymous with Web design, which it is not:

> Editors do make creating web pages as easy as word processing, but they don’t provide students with the skills needed to understand fully what makes web pages work. Also, if students know the basics of HTML, they will find that they have

> the knowledge to debug web page problems they might encounter. By learning HTML without the help of an editor, students are actually expanding their problem-solving skills and learning how to analyze information and develop viable solutions. (Braun, 2000, p. 28)

Teaching Web design solely as software training is ultimately a disservice to students because software like "DreamWeaver does not tell you how to create a web page any more than a French dictionary teaches you to speak French" (Storey, July 2013, para. 2).

Self-Instruction of Web design

Further complicating (educational) matters is the fact that both Web editing software and coding are seen as commonplace—they have begun to merge with computer literacy expectations (Hofstetter, 2006). In other words, Web page creation is slowly joining word processing, emailing, or Internet study habits, for example, as a prerequisite for completing assignments, not an addressed subject. To compensate for this expectation, college students continue the practice of self-instruction begun in the 1990s, for HTML a well as other computer literacies (Carter, 2006). Web design's precedence of self-instruction occurred simply out of necessity: not only students, but "many teachers of Web design are still learning Web design themselves, or are self-taught" (Karper, 2004, p. 162). Web designers routinely teach themselves, becoming experts via experimentation and discourse (Deek, Coppola, Elliot, & O'Daniel, 2000; Karper, 2004; Kotamraju, 1999): "Maybe twenty years from now the [Web] design community will be dominated by the products of college degree programs—but right now, it's still largely dominated by self-taught professionals" (Tuck, 2011, para. 7).

From this self-taught expertise also arose the exponentially-expanding number of instructional materials and textbooks for Web design practices and Web editing software: "As a reflection of this demand, there are many how-to manuals for building Web pages in bookstores and on websites" (Ariga & Watanabe, 2008, p. 815; see also Clark, Knupfer, Mahoney, Kramer, Ghazali, & Al-Ani, 1997; Wang & McKim, 2013). These texts reflect the experts' own experiences of self-instruction and are saturated with assumptions that students can teach themselves to create Web pages, too.

Statement of the Problem

Consequently, there is now the educational assumption that university students can either teach themselves to build Web pages or will encounter Web design instruction elsewhere, as part of software training, perhaps. (This assumption exists for many computer literacies, not just Web design (Kalman & Ellis, 2007).) At the same time, instructors encourage, sometimes require, students to create Web pages or post content to Web pages (like presentations or writings) as assignments or assessment. For example, it has been embraced as a constructivist activity in education programs, used in the service of learning other topics like technology integration in teaching (Bransford, Brown, & Cocking, 2000; Hofstetter, 2006; Koehler & Mishra, 2005; Leahy & Twomey, 2005) or even learning theory (Lim, Plucker, & Bichelmeyer, 2003). Some computer science courses even expect students to complete Web development or programming projects for assignments despite the department not teaching these topics (Stepp et al., 2009). This scenario is detrimental to both students and instructors:

> Far too often it is *assumed* that students will somehow already possess key enabling skills (e.g., study skills, public speaking skills, graphic design skills,

> group management skills)—with the unfortunate results that cause more educators to complain about the absence of those skills than to target them in their planning. Helping students to "learn how to learn" and "how to perform" is both a vital mission and a commonly overlooked one. (Wiggins & McTighe, 2005, p. 59)

It has also led to a dearth of research-based exploration of Web design: "Though the Web is a major contributor to the phenomenon of the Information Age, we know less about it than one might expect…the process by which the Web is produced remains understudied" (Kotamraju, 1999, p. 465; see also Connolly, 2012; Karper, 2004; Park, Saxena, Jagannath, Wiedenbeck, & Forte, 2013).

Of course, there have been prior studies and extensive instructional design, both practitioner and research-based, devoted to coding practices and Web editing software. (The instructional design is evidenced by the plethora of textbooks, instructional websites and professional development options devoted to the topic: A keyword search for "web design textbook" on Amazon.com returns 1,260 results on January 30, 2014.) But, this instructional design most often identifies WHAT should be taught (which content must be covered in order for someone to procedurally create a Web page, e.g.) with only minimal research devoted to HOW it should be taught, so that actual understanding is achieved in a learner-centered environment.

Relying on the current spate of instructional materials perpetuates Wiggins & McTighe's (2005) "'twin sins' of typical instructional design in schools: activity-focused teaching and coverage-focused teaching" (p. 3). In activity-focused teaching, the procedural part of the assignment—building the Web page using software functionality, e.g.—is seen as evidence of learning. The meaning behind the actions is rarely addressed or assessed. Likewise, content-coverage teaching over-relies on the textbook, "allowing it to define the content and sequence of instruction" (Wiggins & McTighe, 2005, p. 21). For example, some Web design courses fail to require students to submit their Web pages for assessment, using the students' knowledge of the textbook or instructional materials as proof of understanding. Both practices are counter to recent advances in instructional theory:

> More than ever, the sheer magnitude of human knowledge renders its coverage by education an impossibility; rather, the goal of education is better conceived as helping students develop the intellectual tools and learning strategies needed to acquire the knowledge that allows people to think productively…" (Bransford et al., 2000, p. 5)

This "new theory of learning" (Bransford et al., 2000, p. 3) in no way rejects the need for students to learn facts in a knowledge-centered environment. Instead, it adds to the standard, by advocating instruction that is also learner-centered, assessment-centered, and community-centered.

Textbook or website-centered Web design curricula have not well-addressed these standards, in most cases. In many instances, they have not even addressed the needs of novices: the curricula often suffer from the instructor's or textbook author's Expert Blind Spot (Bransford et al., 2000; Wiggins & McTighe, 2005), presenting information in a technology-centric manner that only makes sense to other technology experts. (This is not to say that students cannot learn advanced coding and problem-solving using existing materials, or that Web design instructors are not successfully developing their own curricula incorporating these materials—both are possible. There is also a subgenre of

Web design reference books which claims to target the needs of non-technical learners—an example is Robson & Freeman's *Head first HTML and CSS* (2012), which even asks "Tired of reading HTML books that only make sense after you're an expert?" (p. back cover).) These textbooks' significant strength is endorsement of active learning, which is critical for learning to occur: "New developments in the science of learning also emphasize the importance of helping people take control of their own learning" (Bransford et al., 2000, p. 12).

The challenge then becomes to adapt and improve current curricula, by expanding the knowledge base on Web design instruction with "use-driven strategic research and development focused on issues of improving classroom learning and teaching" (Bransford et al., 2000, p. 250), specifically, by developing and evaluating new educational materials and assessment methods for Web design instruction, methods that strive to "teach and measure deep understanding" (Bransford et al., 2000, p. 256).

Project Background

Investigation of this topic began with an autoethnography—a study of my own culture—of the instructional technology (IT) program at the University of Virginia's Curry School of Education. While serving as teaching assistant in a course titled Computer Courseware Tools, I conducted brief lectures, classroom observations, as well as interviews during tutoring sessions, trying to determine why the students experienced difficulty when learning how to build Web pages. (The participants were adult learners, graduate-level education majors, only minimally computer literate, and from primarily non-technical backgrounds, i.e., K-12 teachers.) I volunteered as teaching assistant because I had/have worked professionally in higher education as a Web designer and developer since 1997 and was expert in both hand-coding practices and Web editing software.) A picture gradually emerged of Web design instruction as being overly complex, with too many variables, lacking assessment and feedback, and with little or no emphasis on the Web design and computer language coding rules, guidelines and design strategies most-used by experts.

The project then became the design and development of a concept-based curriculum unit for non-technical learners, for learners that *could not or would not self-instruct*. In particular, the unit strove to address the area that proved most difficult for students during pilotings: Cascading Style Sheets. (See Table 1 for categories of persistent student misunderstandings documented during lesson plan piloting and development.) While students still made accuracy or decision-making errors when creating HTML, CSS was overwhelmingly more difficult for students to comprehend and execute—again, because it entails creating visual design with the notational code of a computer language, an issue of learning how to create multiple representations. Even the most motivated students, or those with prior HTML experience, found CSS's complexity prohibitive, even though CSS "is less intimidating to look at than HTML [and] CSS documents are both simpler in appearance and use a more natural-sounding vocabulary" (Gordon, 2005, p. 66). The onus for some of this difficulty with CSS is again attributable to Internet browser instability; recall that "not all browsers contain support for the latest CSS" (Hofstetter, 2006, p. 290). Still, browser issues do not detract from CSS's conceptual difficulty or the fact that the quantity of CSS terminology is exponentially greater than HTML's (Lie & Saarela, 1999)—both add difficulty during instruction.

Table 1

Student Misunderstanding Categories Compiled during Pilotings – Unabridged

Misunderstood Concept	
CSS styles cascade from outer rules to inner/lower rules	CSS rules that share properties should be grouped, not duplicated
CSS only styles content present in the HTML	CSS properties should not needlessly duplicate default styles
CSS translates to display	CSS properties or values should not be duplicated for the same rule
CSS properties have specific functions and pair with specific values	OS-dependent fonts or colors should not be used
CSS syntax has punctuation and grammar	CSS shorthand has specific punctuation and grammar
Style and content are separate (CSS vs. HTML)	HTML tags may have default styles
CSS layout is created using float, width, clear, display, margin and padding	CSS layout and design must meet accessibility standards
HTML tags translate to display	HTML declaration statements impact browser display
DIV/span are only used when existing tags are insufficient	ID and CLASS describe and differentiate tags
Tags must nest properly	Difficulty editing template
Pathing to URLs, files, and images must be exact	File name conventions must be followed
Tags are added semantically, based on meaning	Deprecated or non-existent HTML tags should not be used
HTML tags have syntax, punctuation, pairing and mirroring	HTML special characters should be used in the code
HTML coding must be accurate and precise	CSS coding must be accurate and precise

Curriculum unit development began with adaption of existing Computer Courseware Tools assignments coupled with an extensive document review of HTML/CSS textbooks and reference books (chosen from Safari Books Online, a service which collects professionally-written technology reference and instructional books). Several other introductory Web design courses or workshops were also observed and analyzed, one offered by the University of Virginia to faculty and staff as professional development, for example. The learning modules and materials for an online Web design course offered by the University of Florida were also reviewed for self-instruction strategies as well as content and activities, as were the W3C's Web Education Community Group curriculum and online courses from the HTML Writers Guild (http://hwg.org/), a W3C member. Interviews with four Web design instructors were also conducted during curriculum unit development, including one with an instructor who specifically taught CSS to non-technical learners (university librarians). The unit was then fully-developed using the ASSURE model of lesson planning found in *Instructional Technology and Media for Learning* (Smaldino, Russell, Heinich, & Molenda, 2005). The ASSURE model was chosen for its constructivistic "heavy emphasis on *active* student engagement in learning activities" (p. 47) and its focus on "planning around the actual classroom use of media and technology" (p. 49).

Development continued through multiple cycles of piloting/implementation, revision based on the outcomes and feedback of the 35 cumulative students who participated, submission to subject matter experts (SME) on both Web design and instructional design, and then revisions based on SME feedback. Significant shifts were made throughout to content sequencing and learning materials—e.g., Web-based instructional videos, tutorials and reference materials were recombined and integrated to provide multiple representations that met the needs of different learning styles (Ainsworth, 1999) and because pilot students rejected one textbook after another as too complex or written for experts. Instructional and assessment methods were also adapted or revised as the unit grew to incorporate the design tenets of the "new theory of learning" (Bransford et al., 2000, p. 3); learner-centered, knowledge-centered, assessment-centered and community-centered elements were integrated into the lesson plans and the learning environment which they created. Also, formative assessment feedback measures took the form of line-item review of students' assignment code, in stark contrast to typical CSS instructional methods, which often lacked feedback altogether. (Formative assessment also included concept-based discussion responses, which were addressed during review sessions.) Supporting instruction on how to study Web-based materials was also added, as it was discovered that the majority of students lacked experience studying Web-based videos, tutorials or tools.

The unit was also evaluated against and then revised based on the "backward design" recommendations of Grant Wiggins and Jay McTighe in *Understanding by Design* (2005), again striving to craft understanding and transfer, not just information coverage:

> Our lessons, units, and courses should be logically inferred from the results sought, not derived from the methods, books, and activities with which we are most comfortable. Curriculum should lay out the most effective ways of achieving results. (Wiggins & McTighe, 2005, p. 14)

In particular, an emphasis on authentic performance stemming from ill-structured, authentic problems and continuous formative assessment was adopted. Finally, a pattern of review/discussion, direct instruction/lecture, demonstration, and then practice solidified in the curriculum unit. Assignments continued the ‘practice’ portion of each lesson, using Web-based instructional materials like videos and tutorials to aid students’ self-directed learning outside the classroom.

Also, while the curriculum unit was most-shaped by constructivist learning theory (because of its tenets on active learning), information processing and behaviorist learning theory strategies were also adopted. For example, because most students had no prior knowledge of computer languages, behaviorist teaching strategies were emphasized through the direct instruction (lecture) and demonstration portions. (Computer literacy studies have shown that technology novices prefer lecture and demonstration when encountering novel information (Kalman & Ellis, 2007).) Information processing ideas, particularly Cognitive Load Theory teaching strategies, were also incorporated for the multimedia or technology learning portions; whenever possible, attempts were made to reduce the ‘noise’ created by interaction with both the Internet and technology (software). The lessons were also structured so that each retrieves and builds on the prior lesson, following the pattern of cognitive schema building (Feldon, 2007).

During the pilotings, students’ misunderstandings and errors were compiled and a list of 28 misunderstanding categories was developed for use in revising the curriculum unit (see Table 1). Though the error frequencies differed by implementation, each of these misunderstandings was consistently persistent throughout the pilotings. The misunderstandings were also present regardless of student’s prior experiences —they represented general conceptual difficulties. They were also not dependent on the particular technology, tool or instructional resource used in a piloting. Regardless of whether the pilot students were from K-12 teaching backgrounds or from information technology backgrounds, for example, they displayed these misunderstandings while learning Web design. (For additional information on classifying HTML and CSS errors, see the findings of Park et al., 2013.)

The curriculum unit's multiple limitations should be noted: to make it both focused and finite, software, graphic design and scripting/programming instruction were excluded, though this reduces its 'authenticity.' Web editing software (Dreamweaver, e.g.) was still used by both students and instructor, but introduction to the software as a coding interface was moved before the unit into an "assignment 0," so that it did not convolute students' concept-based learning of CSS; in other words, students are still required to hand-code, but they can use the software to do so. The need for graphic design instruction was circumvented with the use of downloaded CSS templates, which included Web-ready graphics. And, while the lessons were designed (and materials chosen) to accommodate multiple learning styles, the curriculum unit's Web-based and technology-heavy nature still placed significant technological and cognitive demand on learners.

Purpose of the Study & Rationale

Having addressed one call to action from *How People Learn* (Bransford et al., 2000)—the design and development of a curriculum unit that strives to meet the standards of the new theory of learning—another was undertaken: rigorous analysis and evaluation of the curriculum unit. This study intended to expand the knowledge-base on Web design instruction by examining the viability and validity of the curriculum unit and instructional strategies chosen, as well as adding a level of detail specifically for HTML and CSS instruction.

> It is imperative that we research Web design as its own process so that we can understand how it is changing and impacting our theory, practice, and praxis as well as how it changes our concept of working with existing media and theorizing media yet to come. (Karper, 2004, p. 36)

The study's purpose was also to better document which student conceptual misunderstandings persist and why, particularly for Cascading Style Sheets. (It was a continuation and evolution of the research conducted during the curriculum unit pilotings.) How CSS is best-learned is not well-researched, nor has there been detailed consideration of why CSS is difficult to learn. This is possibly because of the subjectivity often associated with visual style, which CSS mediates (Ariga & Watanabe, 2008; Beriswill, 2005; Taylor, Salces, & Duffy, 2005; Victor, 2002). CSS's conceptual and procedural complexity, the lack of support by Internet browsers, and the overreliance on Web-editing software have also likely served as deterrents to thorough investigation. (To date, there are few education studies even covertly focusing on CSS.) But, CSS is now requisite in Web design; it should be requisite in Web design instruction as well (Gordon, 2005; Maddux et al, 2008).

This study used as its rationale *How People Learn*'s call to bridge educational research and practice through

> use-driven strategic research and development focused on issues of improving classroom learning and teaching. The facts that schools and classrooms are the focus and that enhanced practice and learning are the desired goals render the

> program of research no less important with respect to advancing the theoretical base for how people learn. (Bransford et al., 2000, p. 250)

Research Questions

The research goal driving this study was to contribute to the knowledge-base on Web design instruction, in particular to examine Web design and Cascading Style Sheet instruction, by investigating the viability of the curriculum unit. To that end, the following questions frame this study:

1. What deviations by the instructor occur during implementation of the curriculum unit and why?
2. What student 'misunderstandings' about XHTML and CSS persist throughout curriculum unit implementation and why?

Methodology Overview

In order to evaluate the curriculum unit and examine the research questions, this study used a mixed-method approach and multiple data sources. First, to establish content validity and the appropriateness of the curriculum unit for participants (both instructors and students) an external review panel of experts was assembled. Second, the curriculum unit was implemented at the University of Virginia, with me serving as instructor and participant-observer. An observation protocol was used by both the researcher and trained independent observers to collect data on the implementation. (Both qualitative and quantitative analysis of the observations were conducted.) Third, document analysis of student data (their formative assessments) was completed to support study reliability and the conclusions drawn from the observations.

A Design-based Research (DBR) framework was chosen as most apposite for the study: DBR is grounded in both theory and real-world context and in it, "researchers assume the functions of both designers and researchers, drawing on procedures and methods from both fields" (Wang & Hannafin, 2005, p. 6).

Definition of Key Terms

Cascading Style Sheets (CSS). CSS is a computer language used in Web design in conjunction with Web page markup languages (e.g., XHTML). CSS styles a Web page's content, including visual presentation and layout, for display in Internet browsers (Lie & Bos, 1999). (Technically CSS can be used for print media or other software-based design, but these scenarios are not addressed in this study.)

Push-button publishing. Using a Web-based interface (a website) to post or upload Web content or create Web pages. The term was first used in 1999 by Blogger.com, but genericide quickly occurred, and the term became synonymous with any service that allowed Internet users to publish Web pages or Web content with the push of a button (Howells-Mead, 2009). Examples include blogs and microblogs (Twitter.com), social networking (Facebook.com), photo sharing (Flickr.com), video sharing (Youtube.com), newspaper and journal publishing, as well as content management systems (CMS) and learning management systems (LMS).

Semantic Web design. In semantic Web design, markup code is added based on meaning. The meaning is determined by the Web page content (text, image, script, etc.). For example, a paragraph of text is placed within paragraph HTML tags (<p></p>), not within list tags. Also, style code (CSS) is separated from content markup code (HTML).

Web usability. Refers to how easy user interfaces are to use (Nielsen, 2003). Website usability testing asks questions like "Can users locate information and functionality that they seek on a Web page?" and "Is the Web page free from errors?".

W3C. The World Wide Web Consortium (W3C) refers to the international community of Web professionals that works to develop standards for accessible Web design, communication and computer language usage.

WYSIWYG. Acronym for 'what you see is what you get.' The term is used to categorize Web editors (software) that visually-render code, previewing how the HTML, CSS, graphics, etc. will display in Internet browser software.

XHTML. Extensible Hypertext Markup Language is an updated version of HTML, the language/code used to create Web pages. It includes all HTML tags, but it has additional, stricter, standards-based coding rules for producing Web pages that validate (Hofstetter, 2006). Languages like XHTML are used to "mark up" content like text and images, so that it displays in Internet browsers (W3C.org, 2012).

Summary

Web design as a practice has quickly and steadily increased in complexity, and Web design instruction has followed suit. Investigation and documentation of how Web design, particularly CSS, should be taught to non-technical learners is needed. If Web design is to perpetuate as a 'general' computer literacy (Ariga & Watanabe, 2008), then CSS, and its complex conceptual nature, must be better understood.

Chapter Two

Literature Review

"As educators in a digital age we have a responsibility to provide students with the opportunity to engage with the web in all of its complexity." (Mackey & Ho, 2005, p. 554)

Introduction

This review provides a summary of prior literature addressing Web design instruction, including the myriad of ways in which it is situated and taught in higher education. It also offers an analysis and description of the instructional strategies advocated for both Web design instruction in particular and student-centered instruction in general, including the learning theories that guide the study. (These were the theories and strategies used to design and develop the curriculum unit which this study now strives to evaluate.) The review also incorporates the complexities of teaching and learning technology. For finitude, it does not address the eclectic spectrum of studies debating or codifying how to use the Web in education. Because the prior literature is often from disparate academic fields, with very different foci, it becomes necessary to rely only on a broad definition of Web design:

> Web Design: A multidisciplinary pursuit pertaining to the planning and production of Web sites, including, but not limited to, technical development, information structure, visual design, and networked delivery. (Powell, 2002, p. 15)

Web design is seen by most fields as not just a technological procedure: "a web site, as a product of meaning making, creativity, and problem solving/programming, is a recognizable mode of communication" (Deek et al., 2000, p. 49). More esoterically, "a web design is not only what users see, it is also how they find the information they are looking for, and how they feel about the whole experience" (Burch, 2001, p. 363).

In addition to the term Web design, the various fields employ phrases such as Web composition (Karper, 2004; Turnley, 2005), Web authoring (Niess, Lee, & Kajder, 2008; Turnley, 2005), Web literacy (Karper, 2004; Mackey & Ho, 2005; Maddux et al., 2008), Internet literacy (Hofstetter, 2006), Web development (Blackwell, 2002; Park & Wiedenbeck, 2011), Web engineering (Whitehead, 2002), hypermedia design (Lim et al., 2003), and hypertext authoring (Gordon, 2005)—all ultimately refer to the creation and posting of Web pages.

Studying Web Design

Studying Web design through the lens of education is itself a complex activity. The variability in terminology serves as only the first ambiguity revealed by the literature. Web design is again not only a computer/technological activity, but also a creative/composition activity, which can lead to vastly different trains of thought when it is discussed. The artistic or creative elements associated with the Web design process contribute especially to the difficulty in studying it as an educational topic, since design is often presented as a subjective, reflective, ill-structured action that is difficult to describe, teach and establish evidence for (Ariga & Watanabe, 2008; Beriswill, 2005; Clark et al., 1997; Taylor et al., 2005; Victor, 2002).

There is further “paucity and inaccessibility of data” (Kotamraju, 1999, p. 465) when studying Web design, simply because the Web and related technology evolve so rapidly:

> Digital technologies such as the World Wide Web diffuse through populations during, not just after, the modification process…digital technology diffuses more quickly than other technologies, mainly because diffusion’s beasts of burden, transportation and communication, are faster and more efficient. (Kotamraju, 1999, p. 467)

Web design as practice exists in this constant state of modification or “maintenance” (Taylor et al., 2005), and establishing evidence and conclusions are difficult, since the technology (evidence) can be “written over, erased, replaced, and forgotten with ease, speed, and low cost” (Kotamraju, 1999, p. 467); “Digital technologies move so quickly that it is a challenge to capture them once, much less to repeat the process” (Kotamraju, 1999, p. 471). Instructional materials for Web design and development are also rarely able to keep pace with the Internet’s rapid changes, which has led to considerable amounts of outdated and incorrect reference books or websites still being in circulation (Stepp et al., 2009; Storey, September 2013).

This ever-shifting, time-compressed scenario has led to hesitancy by academia to pursue in-depth exploration of a research topic with such instability (Connolly, 2012; Kotamraju, 1999; Park et al., 2013; Victor, 2002). For example:

> What we do not have is a large variety of scholarship…that discuss[es] Web design as a process, and more specifically as a composing or rhetorical process. It

> is my belief that this lack is due to the inability to keep pace with the rapidly changing nature of the Web. (Karper, 2004, p. 6)

Academia has also resisted studying Web design education by classifying it as a skill or literacy (Karper, 2004; Royal, 2005) and then relegating it as skills-based training (Chafy, 1997). In the latter half of the 20th century, educational research strove to emphasize the teaching of critical thinking, expression and application of knowledge, rather than skills labeled 'basic literacies' (Bransford et al., 2000) or professional or procedural training (Chafy, 1997). This meant that studying an interdisciplinary educational topic like the World Wide Web (where technology, communications, and design topics intermingle) was considered "just beyond the scope of both the technical and the non-technical disciplines, both of which remain[ed] largely content to focus on skill-based education" when teaching technology (Chafy, 1997, p. 17). In other words, because Web design software quickly emerged, education often grouped Web design instruction with other software or computer literacy training (Braun, 2000). Investigation of the topic only occasionally extended further—recall Karper's (2004) comment: "What we do not have is a large variety of scholarship…that discuss[es] Web design as a process, and more specifically as a composing or rhetorical process" (p. 6). Hannafin & Kim (2003) extend this criticism of a lack of educational scholarship on Web design to teaching and learning with the Web in general:

> We have, for the most part, failed to break much new pedagogical ground with our collective effort to date (p. 347)…where research focusing on the Web's unique affordances has been conducted, it is too diffuse and unfocused to generate meaningful guidelines…[it] repeats the mistakes of researchers in other fields or

> arrives at conclusions long-since accepted by researchers and practitioners outside their field. It is inefficient and misleading. (p. 349)

This conclusion rings true for Web design research particularly, since much instructional design research devoted to the topic has been conducted by practitioners or professionals (Karper, 2004), and they have already reached conclusions about teaching Web design which academia has not even investigated:

> It is imperative that we research Web design as its own process so that we can understand how it is changing and impacting our theory, practice, and praxis as well as how it changes our concept of working with existing media and theorizing media yet to come (p. 36)…Such research would help to enhance and legitimize the teaching of Web design. (Karper, 2004, p. 21)

More generally, "researchers need to embrace technological innovation and continually re-examine how innovation alters the definition of learning and instruction, and how we study them" (Lawless & Brown, 2003, p. 229).

Advocacy for Teaching Web Design

The lack of scholarly research on Web design instruction is countered by an ample range of advocacy for teaching the topic, particularly at the higher-education level (e.g., Hofstetter, 2006; Karper, 2004; Mackey, 2005; Rosmaita, 2006). As example:

> The explosive growth of the Internet has made the knowledge and skills for creating Web pages into general subjects that all students should learn. (Ariga & Watanabe, 2008, p. 815)

> It has become imperative at the college level to have a course taught in the area of HTML, HyperText Markup Language and web-design. (Mull, 2001, p. 1)

Statements like these permeate Web design literature, likely as a way to transition the topic beyond a ‘software training’ stigma (Walker, 2002) and to encourage use of Web design as a student-centered learning activity (Hofstetter, 2006; Lim et al., 2003; Mackey & Ho, 2005; Niess et al., 2008). Technology-centric fields such as instructional technology or engineering sometimes go so far as to advocate entire programs of study or even a new academic field to address Web design (Hadjerrouit, 2005; Krunić et al., 2006; Victor, 2002; Wang & McKim, 2013; Whitehead, 2002). It is also seen as an increasingly essential component in information literacy education (Mackey & Ho, 2005).

Many educators still label Web design instruction as a skill or training though (e.g., Hofstetter, 2006; Leahy & Twomey, 2005; Mackey, 2005), but they simultaneously argue for the importance of skills instruction in an academic setting: “skill-based education has become critical to the survival of all disciplines, and the closer a program of study is allied to servicing the needs of our technology-driven society, the better” (Chafy, 1997, p. 17; see also Mull, 2001). Chafy’s (1997) comment is only one attitude in an expansive, ambiguous debate about the role of skills instruction in higher education, but it should be noted, because this ‘academic or professional’ debate envelopes Web design instruction and the scholarship focusing on it: “The website design class…becomes a ‘contact zone’ (Pratt) because the worlds of the academic and professional are intersecting. The instructional environment is academic, but the Internet community in which students are participating is professional” (Walker, 2002, p. 66). Web design educators typically acknowledge that a University “must balance the practical application of knowledge in real-world settings with the ability to critique events" (Royal, 2005, p. 402), i.e., with critical thinking or problem-solving competencies (Bransford et al., 2000; Victor, 2002). Many also recognize that

> the ultimate goal of schooling is to help students transfer what they have learned in school to everyday settings of home, community, and workplace…an important strategy for enhancing transfer from schools to other settings may be to better understand the nonschool environments in which students must function. (Bransford et al., 2000, p. 73)

The identification of Web design as a valued professional activity is one reason why its instruction is advocated (Mull, 2001): "Simply put, students really want to learn this stuff" (Stepp et al., 2009). Karper (2004) reports in her study of English majors learning Web design that

> participants emphasized their desire to learn about Web page design due to pressing professional concerns (either pedagogical or job-related)…participant focus was on primarily creating pages to be used in teaching and learning, for marketing themselves as viable job candidates, and for other professional reasons. (p. 88)

This student motivation makes Web design an appealing activity to educators, not only because they see it as an 'authentic' experience that students will take seriously (Karper, 2004; Marx, 2003), but because

> research suggests that in a student-as-hypermedia designer approach, students are highly satisfied with the activities, develop skills and knowledge effectively, are mentally engaged to a much greater extent by developing materials than by studying materials, are highly motivated by the activity because they gain a sense

> of ownership in the product and in their learning, and are actively engaged in creating representations of their own modes of expression. (Lim et al., 2003, p. 14)

Its appeal to educators as a student-centered activity repositions Web design as more than just desirable professional knowledge; it is valuable because it promotes student engagement, ownership of learning, and the knowledge to create, not just consume, technology (Hofstetter, 2006). This contributes to Web design's use as a constructivist learning activity, used in the service of learning additional content like technology integration in teaching (Bransford et al., 2000; Hofstetter, 2006; Koehler & Mishra, 2005; Leahy & Twomey, 2005), learning theory (Lim et al., 2003), rhetoric and composition (Karper, 2004) or writing and communication (Mackey & Ho, 2005), to list but a few: "Web design is becoming a crucial component of pedagogy, even in classes that are neither distance education nor meeting full-time in computer labs" (Karper, 2004, p. 22).

Advocates for teaching Web design also argue that "the skills required for *information production* are as valuable as those skills required for information access and evaluation… research and production abilities are inter-related and should not be separated in our teaching practices" (Mackey, 2005, p. 3241). (By 'production ability', they refer to building Web pages.) They see learning Web design as part of a "larger pedagogical context" (Mackey & Ho, 2005, p. 543), one with advanced, desirable learning outcomes:

> When students produce original documents for the web, they must understand how to evaluate a range of sources, how to properly document all sources of

> information, how to work ethically in digital environments, and how to participate in a collaborative process with other developers and users. This active-learning approach moves beyond discrete computer skills and toward critical thinking and evaluation, as well as writing and communication, all of which are essential and inter-related [Information Literacy] skills. (Mackey & Ho, 2005, p. 543)

> The very nature of the activities involved in Web authoring involves problem solving and decision making. Engaging students in designing Web pages along with solving problems in a particular content area provides them with powerful experiences with problem solving and decision making. (Niess et al., 2008, p. 208)

Learning Web design also offers students “a set of information and technology skills that advance research ability and critical thinking while developing proficiency in the production and publication of original content” (Mackey, 2005, p. 3240)—much sought-after skills at the higher-education level (Bransford et al., 2000).

Web Design as Information Literacy

Though Web design instruction is often advocated in support of other subjects—learning Web design while also learning how to teach with technology, for example—computer literacy instruction has enfolded it as a singular topic (Carter, 2006; Kalman & Ellis, 2007). Web design “is transitioning from a cutting-edge technology to a more accepted, more transparent technology” (Karper, 2004, p. 35), because of the Internet’s presence in daily life; this sometimes means that computer literacy instruction presents Web design as software training, but more often these educators reclassify it as an advanced ‘new literacy’ that requires integration of not just technological skills, but the judgment and perspective of communication and composition skills in the Web medium (Karper, 2004; Krunić et al., 2006; Mackey & Ho, 2005; Royal, 2005; Turnley, 2005). In other words, Web design is classified as an information literacy—Mackey and Ho (2005) summarize the viewpoint:

> Information literacy focuses on content and communication: it encompasses authoring, information finding and organization, research, and information analysis, assessment, and evaluation…the overall emphasis on critical thinking and lifelong learning moves beyond a rudimentary computer literacy model.…In a digital information environment such as the web, it is essential to recognize that content, communication, and technology are inter-related and interconnected. (p. 544)

The definition(s) of computer literacy is itself constantly evolving to accommodate Web design advancements and to move beyond the model of software training—it is evolving into what Mackey and Ho (2005) refer to as a "convergent model" of computer literacy, in which "people engage with technology to actively create and produce, rather than …simply use a particular software program or hardware device" (p. 544): "Rather than letting hardware and software drive web-based learning activities, instruction should highlight contextualized technology use and articulate the cultural and rhetorical positionings of the Web" (Turnley, 2005, p. 133). Royal (2005) also argues for this theoretical evolution of computer literacy:

> When classes [focus solely on technical skill], it is difficult for a curriculum to integrate learning across skills, programs, or platforms. While this approach might be best for developing highly skilled technicians in particular areas, it lacks a

> perspective on the new media environment that can influence not only design decisions, but can offer critical and theoretical understanding of multimedia issues. (p. 401)

It should be noted that many traditional computer literacy standards promote advanced intellectual skills, including problem solving and design, as "critical elements in technology education" (Jakovljevic, Ankiewicz, de Swardt, & Gross, 2004, p. 267), even while focusing on software or hardware training—these two ideas are not mutually exclusive (Ebersole, 1997; Niess et al., 2008). Accreditation standards for colleges and universities also mandate the integration of information literacy and technology as intellectual skills in higher education curricula rather than just software training (Mackey & Ho, 2005).

Web design instruction then becomes a path to this new digital information literacy, as well as to evolving computer literacy, because "Web composing blurs the lines of our discipline and makes it 'difficult to tell when literacy ends and technological proficiency begins' (Lassota Bauman, 1999, p. 270)" (Turnley, 2005, p. 133). For example, "when students create hypermedia products [i.e., Web pages], the challenge is to search for information, arrange and organize the information appropriately, and explore relationships among pieces of information in new and different ways" (Lim et al., 2003, p. 13). What this description leaves out is that these "new and different ways" include both the visual design and the nonsequential design of information that is the nature of Web pages—hypermedia are "nonsequential documents containing not only text, but also elements such as audio, video, graphics, drawings, photographs, and animation, along with computer systems on which these components are stored and displayed" (Niess et al., 2008, p. 187). This scenario is a reminder that even when reclassified as an information literacy, Web design cannot separate from its technological requirements.

Employing Web design instruction and assignments to teach a combination of computer and information literacies can lead to difficulty in balancing the two areas. Turnley (2005) offers this evidence:

> …even when instructors strive to foreground rhetorical frameworks, students may privilege proficiency with technical tools over issues such as audience, purpose, and argument. Such instrumentalist notions of technological literacy are a challenge to all instructors who teach critical approaches to web authoring. (p. 131)

Students are "more interested in learning Web skills than in learning content applicable to the course" (Lim et al., 2003, p. 17), because of the professional value attributed to technological skills (Karper, 2004; Mull, 2001). They may also find the computer literacy portions of instruction (e.g., the procedural process of building a Web page using HTML) less complex to grasp than the conceptual nature of the information literacy portions of the lesson. For example, Karper (2004) writes: "I had expected to find that my designers would be most frustrated with technology-related issues, and that technology would be the only mediating and/or complicating factor…they all identified rhetorical rather than technological concepts...layout, design, purpose, audience…" (p. 108). (To clarify, by "technology-related issues" Karper refers to software or hardware difficulty; she labels information literacy difficulties as 'rhetorical'.) Leahy and Twomey (2005) report similar findings—"Technical problems were mostly associated with the design, construction and editing of links, or with inserting and aligning graphics" (p. 147)—but as can be seen, they label the same 'design' difficulties as technical rather than informational. (Terminology confusions or ambiguities of this nature are typical when Web design instruction is discussed.) Regardless, the relevant point is that teaching Web design as a combination of literacies only adds to its complexity as an educational topic.

The term 'Web literacy' is sometimes used now instead of information or computer literacy to categorize Web design, usually in technology-centric fields like computer literacy or engineering. As with the term 'Web design', definitions of Web literacy vary by field, if not author. They typically emphasize "the evaluation and/or production of web information" (Mackey & Ho, 2005, p. 546), and include "a set of skills in web development knowledge (producing documents in HTML, XHMTL, XML, and CSS), and web environment knowledge (web usability, web accessibility, information architecture, information ethics)" (Mackey & Ho, 2005, p. 548). These Web design elements have otherwise been assigned to both general computer literacy and information literacy education, and, "whether we designate specific types of literacy or whether we fold the changing expectations of the Web and new media into our existing ideas about literacy, [Web design] clearly remains a key concept" (Karper, 2004, p. 40) because the information and technical competencies it teaches are seen as essential in modern education (Ariga & Watanabe, 2008; Hadjerrouit, 2005; Hofstetter, 2006; Mackey, 2005; Mull, 2001). It does require that educators "develop new, medium-specific strategies and re-imagine functional aspects of computer literacy (Selber, 2004)" (Turnley, 2005, p. 133) to meet Web design's expanded/expanding role (Marx, 2003).

Web Design as Visual Literacy

As if the combination of computer and information literacy demands attached to Web design instruction did not provide enough complexity, there is a third aspect to consider: visual literacy (Clark et al., 1997). Visual design concepts—e.g., use of graphics, color, and layout—are essential for creating Web pages: "Graphic arts skills are very important in determining the graphic design and layout of Web pages, and the graphic design of a site can have a significant impact on its success, and its usability" (Whitehead, 2002, p. 22); "to effectively communicate with visuals, creators of web pages must consider the simplicity and clarity of the images, balance, harmony and organization of the text and images, aspects of framing, and emphasis color, texture, and space (Thompson, 1994)" (Clark et al., 1997, p. 357). Recall also Niess et al.'s (2008) description of Web pages as "nonsequential documents containing not only text, but also elements such as audio, video, graphics, drawings, photographs, and animation" (p. 187). Web design can then be seen as not just information or communication design; it is also the visual design of information (i.e., visual literacy), where Web pages are "a 'new space' for writing, a space in which words [are] not the primary means of communication, but where images, animation, sound, and other forms of media should perhaps be given primacy over written text" (Karper, 2004, p. 45). Some educators go so far as to classify Web design as "a form of art" (Mull, 2001, p. 4), but most do not venture further than labeling its visual design processes as a mixture of artistic and technological aspects (Taylor et al., 2005)—as ever, Web design cannot be divorced from its technological context.

There is seemingly little research attending to teaching the visual aspects of Web design, and when visual design is broached, it is most often textual or technical:

> It is now common to teach the technical side of the production of Web pages and many teaching materials have been developed. However teaching the aesthetic side of Web page design has been neglected (p. 815)…Web page creation courses do not normally take time to consider visual design except in art or design departments...Because visual expression is judged subjectively, it is considered to be unnecessary, difficult, or impossible to teach. However, aesthetic design affects the quality of information on Web pages, and it is apparent that the appropriate artistic elements, color, and layout enhance the visual appeal. (Ariga & Watanabe, 2008, p. 816)

This is not to say that Web design instruction does not routinely include topics like graphics, color, and layout—on the contrary. But, they are presented as part of the technical or procedural requirements for building a Web page (Sklar, 2008; Teague, 2006) and "the myriad of technical manuals…provide information about how to create web pages but they don't specifically address the necessary design elements that will help the web page communicate clearly and appear aesthetically pleasing" (Clark et al., 1997, p. 356). Visual design is viewed as "an abstract process [that] cannot be reduced to sequential procedures or lists of a guideline for adequate design" (Ariga & Watanabe, 2008, p. 817)—this puts it at odds with the procedural processes typically used to teach Web page creation. Teaching visual design as part of Web design instruction is an added complexity, because "the Web is so far reaching in content and design that no collection of [Web] pages represents what is typical" (Sklar, (2008), p. 31) and "there is a little agreement on the inclusion of elements of good screen design, appropriate size of graphics, use of icons for navigational purposes, and designing the screen as a portrait"

(Clark et al., 1997, p. 360) for example, because these elements can differ on any, if not every, Web page.

The lack of a visual-design procedural model or concrete standards that Web page text or layout must meet, for example, is especially problematic for novice students: "Students without practice in graphic design need some guidelines to conceive the visual expression of Web pages; otherwise, they cannot begin to design it at all, or make the visual design heedlessly" (Ariga & Watanabe, 2008, p. 827). This leads to students creating Web pages "decorated with fancy designs that do not match the contents" (Ariga & Watanabe, 2008, p. 817), or to creating Web pages with

> no balance between their function and form…the main reasons for this problem are the web designer's inexperience, short deadlines, and the so-called ad-hoc design without adopting any web design models. We believe that these problems occur due to the fact that web design courses are mostly focused on technologies, programs and scripts… (Krunić et al., 2006, p. 319)

Even students with prior graphic design experience suffer from the lack of visual design instruction specifically for the Web medium. Karper (2004) reports that "print skills did not automatically assist [students] in developing Web skills" (p. 54) during her study. Also, while many students "did fine analyzing Web design and generating principles for aspects of 'good' design, most of the students could not successfully apply these principles to the creation of their own Web-based projects" (Karper, 2004, p. 17). This is contrary to the expectations of some higher education instructors that not only do many college students already possess print design skills (Wiggins & McTighe, 2005), but "as in the case with the print media, people who have talent in designing graphics, have a knack with colors…can do a good job designing successful web sites" (Deek et al., 2000, pp. 48-49). (This may be less of an issue for future students, since graphic and print design are gradually becoming synonymous with Web design (Bureau of Labor Statistics, 2013).)

Visual design instruction is sometimes grouped with Web usability or accessibility topics (Connolly, 2012; Mackey & Ho, 2005) during Web design instruction, and in these instances design guidelines do emerge. For example, students may be taught that "the color of the text must be visible and readable against the background that they chose" (Mull, 2001, p. 23). While useful for effective textual communication on the Web, these guidelines still skirt the teaching of visual literacy:

> Although there are scientific aspects of web page layout, for example avoiding the use of certain colour combinations in order to avoid potential problems for colourblind users, or using high contrast colour variations to cater for partially sighted users, the majority of web page layout design still centres around the artistic aim of producing web pages that are 'visually pleasing'… (Taylor et al., 2005, p. 337)

'Visually pleasing' does not necessitate using graphics or images on Web pages, per se—"using graphics by itself does not necessarily enhance visual expression" (Ariga & Watanabe, 2008, p. 816). And, fields that teach Web design as part of communications or composition studies for example may advocate 'minimalist' graphic design that affords visual lucidity instead: "the purpose of proper web design is to reduce the level of *noise* in Internet communications" (Burch, 2001, p. 360).

This clarity-in-communication approach towards visual design literacy is shared by educators that advocate Web design instruction only as part of learning other non-technical subjects: “emphasis is always on creating relevant and engaging content rather than on complicated design” (Leahy & Twomey, 2005, p. 145). Leahy and Twomey (2005) here identify ‘content’ as the research and writing generated by students during the course and the Web page as simply the mode of presentation, a scenario often employed when Web design is used as a student-centered learning activity (e.g., Hofstetter, 2006; Koehler & Mishra, 2005; Lim et al., 2003). Again, the issue is that the Web and Web pages are a visual medium, but these fields fail to focus on the visual aspects of Web page creation (Ariga & Watanabe, 2008). Also, Niess et al. (2008) argue that “the process for designing and authoring Web pages is similar to that described for movies and presentations” (p. 193), but this is actually counter to Karper’s (2004) findings: “composing processes for Web designers are substantially different from composing processes in other media, a finding which has serious implications for research and teaching” (p. 107).

Technology-centric fields like computer science or engineering typically exclude Web design’s artistic elements, not because of the associated subjectivity or complexity, but because they simply do not identify aesthetics as their purview: “it is uncommon to find graduate-level courses introducing graphic design to people with technical backgrounds” (Whitehead, 2002, p. 23). The ‘science vs. art’ attitude persists in engineering education in particular—“the worlds of science and engineering and that of art and design are two alien cultures” (Taylor et al., 2005, p. 333)—and unlike the overlap visible in many computer and information literacy courses,

> computing courses may sometimes suffer from a lack of appreciation of the benefits of artistic creativity in the design process, and art, media and design courses may sometimes suffer from a lack of student appreciation of the benefits of science and theory in the design process. (Taylor et al., 2005, p. 339)

But, just as the computer and information literacy fields are evolving to accommodate the affordances of the Web (Mackey & Ho, 2005), so engineering evolves to accommodate visual design: "when teaching web site design it is necessary to impart the creative, artistic aspects of design (as well as technical aspects) to computing students" (Taylor et al., 2005, p. 331); "graphic arts seem to have a place within a Web engineering curriculum, and should belong within the set of key knowledge areas" (Whitehead, 2002, p. 23).

Advocacy for teaching Web design as a visual literacy is still minimal when compared to that of computer or information literacy (Mull, 2001). Web design educators may recognize that "visual design is a significant factor in the development of web design" (Clark et al., 1997, p. 356), but the complexity of adding subjective topics like aesthetics discourages practice, because it "poses many challenges that traditional instructors are not accustomed to" (Burch, 2001, p. 362). Krunić et al. (2006) are an exception; they argue for a combination of technology and art education, with a series of courses devoted to Web design's creative processes:

> Students ought to study subject areas such as graphic design, form and style, and drawing and painting (p. 318)…During the second year several specialized courses are added introducing the students into the creative process of making

> websites. Art courses are evenly distributed throughout the first two years of the learning process since web design has its artistic component as well. (p. 325)

This call for a combination of technological and artistic instruction represents an incredibly complex instructional scenario. Still, Taylor et al. (2005) offer evidence that it is feasible:

> The majority [of students] appeared capable of mastering both the technical aspects of utilizing a web development software package and creating technical design documents, and the artistic aspects of creating appropriate visual styles and layouts for the prototype web site created as part of the coursework. (p. 339)

Teaching HTML & Web Editing Software

Regardless of the technological, artistic or literacy focus of the course in which it is taught, most if not all Web design instruction includes procedural knowledge for creating Web pages with the HTML computer language and/or Web editing software (Hofstetter, 2006; Karper, 2004). Hofstetter (2006) summarizes three options:

> First, you can use an HTML editor to create a Web page by working directly with the hypertext markup language, in which all Web pages are encoded. This method provides you with a good understanding of how HTML works, but it is technical. Second, you can use the 'Save the Web Page' option to convert word-processed documents into Web pages…Third, you can use a what-you-see-is-what-you-get (WYSIWYG) editor to create Web pages through a graphical user interface that lets you enter text and graphics directly onto the screen exactly as you want them to appear. As you create the screen, the WYSIWYG editor automatically generates the HTML code that makes the Web page. (p. 208)

The connection to word processing. The second option mentioned above, converting word processing documents, persists despite the W3C and almost all professional Web standards imploring against the practice: "Don't use a word processor, such as Microsoft Word or OpenOffice" Bos writes for the W3C, for example (2004, Step 1: Writing the HTML section, para. 2). Governing organizations such as the W3C argue against word processing conversion to HTML because "many HTML applications have trouble dealing with Word's extra formatting codes that it places in a standard web page document" (Wempen, Chase, Jacobs, McCall, Nielsen, & Schmid, 2006, p. 860)—word processing software were, after all, *not* created for composing Web pages or creating files for Web browsers. But, this faulty use to generate/convert HTML pages persists, likely because college students are expected to already know how to use word processing software and because they are seen as common and easy-to-use (Carter, 2006; Kalman & Ellis, 2007; Karper, 2004; Lever-Duffy, McDonald, & Mizell, 2005): "Although dedicated web authoring tools are easy to use once you have mastered the skills, they are typically not as easy as using a web component of an alternative software package with which you are already familiar" (Lever-Duffy et al., 2005, p. 265). Also, some early Web design instructors supported using word processing software as a way to "deemphasize technology use or find ways to avoid having to teach HTML tagging" (Karper, 2004, p. 13)—these early instructors argued that teaching HTML was prohibitive for non-technical learners or did not want to displace existing content "in order to fit the technology into an already overcrowded semester" (Karper, 2004, p. 14).

The connection between word processing and Web page creation also persists because some Web design instruction relates Web pages to print pages, rather than treating them as a separate media (Gordon, 2005; Karper, 2004). WYSIWIG editors in particular rely on the "print metaphor" (Karper, 2004, p. 159); this can offer 'technological comfort' to students, by allowing them to draw upon their prior experiences with word processing software:

> students often relied on their knowledge of other interfaces to help make sense of the interface at hand [i.e., the Web editing software]. In fact, more than any other variable, the students' ability to make comparisons to other interfaces during composing may have predicted success in managing and completing the heavily procedural Web-composing tasks. (Dick, 2006, p. 212)

Prior experience with word processing software offers no guarantee of successfully learning Web design though, and the divergent technological and visual design demands of the Web design process ultimately overshadow software similarities:

> For some students, the transition from linear, paper documents to nonlinear, hypertextual documents is fairly natural; many college students today are fairly handy with multiple modes of document composition. However, for many others, the leap from Word to Dreamweaver is a major challenge. To many novices HTML looks like a programming language. (Gordon, 2005, p. 58)

Web editing software. Web editing software are still the preferred tool used in most Web design courses, particularly those aimed at non-technical learners: "For anyone more interested in writing text rather than memorizing multiple markup codes, editors with toolbar buttons similar to those seen in word processing programs are certainly the easiest way to format text" (Notess, 2006, p. 44; see also Lever-Duffy et al., 2005). The procedural automation offered by WYSIWYG software "help[s] people create Web pages without particular knowledge or technique. In this circumstance, making Web pages does not require advanced skills" (Ariga & Watanabe, 2008, p. 815). However, "preprogrammed tools in web design software do not afford their users the means to build sites that are fully compatible with users, browsers, and standards" (Voegele, 2006, p. 2). ('Web design software' here is synonymous with a WYSIWYG editor or Web editor.) These software may be "lightweight" and only "provide the most popular choices" (Notess, 2006, p. 44), rather than the choices needed to create standards-compliant Web pages. Likewise,

> there continues to be no such thing as a true *what-you-see-is*-what-*you-get* editor, and it is almost always necessary to 'tweak' the code produced by any of the existing editors. To do so effectively and easily, one must master at least the rudiments of markup. (Maddux et al., 2008, p. 4229)

For this reason, teaching Web design using an HTML editor is frequently advocated as well (Braun, 2000):

> The advantage of creating Web pages with an HTML editor is that it gives you more control over the Web page than WYSIWIG editors and HTML translators, which create the HTML for you. The disadvantage is that for less technically inclined authors, editing HTML tags can seem tedious and time-consuming. (Hofstetter, 2006, p. 208)

In reality, many Web editing software contain both a graphical WYSIWYG interface and an HTML editor for building Web pages (see NVU or Adobe Dreamweaver, for example), making the distinction between them less prevalent as Web design evolves. Which area a Web design course focuses on (if not both), depends on the skills the content area or instructor emphasizes. Regardless, Web editing software have been seen as essential to learning Web design from the beginning, particularly as a way to manage the topic's complexity:

> Web site design skills constantly incorporated new software essential to design…as the technologies of Web site design grow more complex, software to manage that complexity by automating aspects of Web site design emerged and became part of the repertoire of Web site design skill. (Kotamraju, 1999, p. 466)

The issue for students then becomes not just the difficulty of learning Web design, but also the difficulty of learning how to use Web design software (Ariga & Watanabe, 2008). Dick (2006) concludes: "That every student sought help supports the contention that even 'easy-to-use' interfaces are difficult to learn and require a well thought-out, research-grounded pedagogy" (p. 214). Karper (2004) also reports extensively on how "the editors themselves are not novice-friendly" (p. 33) and "often times the technology contributed to [students] procedural confusion" (p. 134) when they built Web pages:

> Many of them complained about the technologies not 'speaking their language' through their interface and online help during interviews or in surveys, and identified that as being a major technological hurdle in page creation (p. 34)…Designers' difficulties with getting technologies to 'translate' their rhetorical choices, as well as how what they could do with the technologies changed their rhetorical choices and composing processes (p. 70)…They also expressed frustrations with being unable to 'make computers do what they wanted'. (Karper, 2004, p. 81)

In spite of the technological difficulty, most students and instructors still welcome the automation or comfort based on familiarity offered by Web editing software (Clark et al., 1997; Descy, 1999; Dick, 2006; Hofstetter, 2006). For example, Liu and Downing (2010) found that their students "were excited to learn they could visually design a fly-out menu in CSS without a single line of code" (p. 276). Mull (2001) also offers an effective strategy for teaching Web design software:

> The programs being taught were of a hands-on nature and, to be learned, had to actually be utilized and executed. Simply lecturing and, then, letting the students use the programs was not enough for them to learn. The students were taught each aspect of a program as they went. Once an aspect was taught it was applied. This let the students learn the programs, and was therefore more effective, then teaching them several operations at once and then letting them apply them an hour after they had heard what to do. This made the class move smoother and allowed for fewer questions. (p. 28)

As described, this is very much a procedural process, like that employed in other software training. Students still learn conceptual information about Web design when interacting with Web editing software, but because of their reliance on procedural rather than conceptual learning during a process like the one Mull describes, students

> develop a strong attachment to the interface of particular programs…and feel that their knowledge about Web design [is] intimately connected either with the Web page editors they were using and/or with the type of computer they used (usually a PC running Windows). (Karper, 2004, p. 139)

Dick (2006) ultimately argues that "both individual personality differences and previous experience play a significant role in how students [learn] to use Web-composing interfaces" (p. 213-214). It has been shown that students can become less tied to a single software, given the opportunity to build their experience:

> Singley and Anderson taught students several text editors, one after another, and sought to predict transfer, defined as the savings in time of learning a new editor when it was not taught first. They found that students learned subsequent text editors more rapidly and that the number of procedural elements shared by two text editors predicted the amount of this transfer. In fact, there was large transfer across editors that were very different in surface structures, but that had common abstract structures. (Bransford et al., 2000, p. 65)

Still, Karper (2004) reports that for many students, transferring the concepts of Web design from one software to another becomes "far too daunting" (p. 139). This is an unfortunate finding, because "those students who can move beyond the procedure of the interface are better able to appreciate more fully the wide range of factors that affect, mediate, and sometimes hinder their technological literacy development" (Dick, 2006, p. 211-212). Students in effect learn not how to create quality Web pages, but how the software *allows* them to create Web pages; "learning the procedures associated with the technologies [becomes] a major component of the evolution of the designers' composing processes" (Karper, 2004, p. 127). For both technology and literacy educators, this represents a danger (Karper, 2004; Kotamraju, 1999; Niess et al., 2008): "When interactions with technologies are assumed to be automatic rather than contingent upon personal, social, and political factors, users (including students and teachers) are positioned as passive receivers rather than active agents" (Turnley, 2005, p. 134). Karper (2004) relates this viewpoint back to teaching Web design software:

> As a technology integrates itself into culture, people stop thinking about it having any influence on writing. Fewer researchers study the effects of word processing software on the writing process since word processing is no longer a novel technology but considered to be an expected and integral part of the writing process for most people (p. 127)…As Batschelet (2004) points out in her discussion of the print metaphor that most WYSIWIG editors use '[in] attempting to make the Web writing process familiar, authoring programs strip away just want students need most: a sense of the demands the new medium will place upon them'. (p. 159)

HTML and coding. One of the most significant 'demands' of teaching and learning Web design is of course the computer languages needed to create Web pages (e.g., HTML, XHTML, CSS, XML). The extent to which HTML and coding practices should be incorporated into Web design instruction is yet another debate without consensus, and "the decision to teach HTML coding as an alternative to using packaged Web-design software has shown mixed results (Mauriello, Pagnucci, & Winner, 1999; Rea & White, 1999)" (Dick, 2006, p. 206). In reality, HTML is usually included in Web design instruction *in addition to* WYSIWIG software, rather than alternative to it (Hofstetter, 2006). (Again, many Web editing software contain both WYSIWIG and HTML editing capabilities.) HTML instruction is sometimes added simply to compensate for a lack of software functionality (Lever-Duffy et al., 2005): "sometimes you do need to work at the level of the HTML code to create a special effect or to insert a command that a certain tool may not yet handle" (Hofstetter, 2006, p. 232). In other instances, the Web editing software generates problematic code, incompatible with different Web browsers, for example—"HTML background knowledge is key in any web design. Many times web editing programs may not produce the right results. The individual must then go back, using HTML, to fix the problem" (Mull, 2001, p. 6). But, most Web design educators advocate HTML's inclusion as a continuation of the argument that Web editing software limits students' conceptual learning of Web design:

> Web editors allow users to create and publish web pages without knowing HTML code. Although this short cut may expedite students' web authoring, it also can serve to mystify technical aspects of the process. Placing technology behind the scenes can decontextualize key aspects of web production and thus limit students' rhetorical agency. Students do not have to become HTML experts in order to create effective web pages, but a basic knowledge of HTML can enhance their practical and conceptual understanding of web-based documents (Gresham, 1999; Mauriello, Pagnucci, & Winner, 1999). (Turnley, 2005, p. 133)

Learning HTML then becomes pedagogically desirable, because WYSIWIG editors

> don't provide students with the skills needed to understand fully what makes web pages work…By learning HTML without the help of an editor, students are actually expanding their problem-solving skills and learning how to analyze information and develop viable solutions. (Braun, 2000, p. 28)

(By 'editor', Braun means WYSIWIG editors that automate HTML code generation, not HTML editing software used for typing and formatting HTML code.) Braun is not alone in this rejection of WYSIWIG editors in favor of teaching HTML coding (Karper, 2004; Maddux et al., 2008; Mull, 2001; Rosmaita, 2006):

> Students need an understanding of [XHTML and CSS] concepts because these skills encompass the foundation of all Web development and design. Even with the advent of Web editing programs, in order for the student to feel he/she has the maximum control over the development environment, one must be able to understand and interpret the underlying code. (Royal, 2005, p. 406)

Working directly with the computer language HTML empowers Web design students by offering them the “knowledge needed to exercise the greatest control possible over the design and function of their works” (Gordon, 2005, p. 68) and by giving them a “chance to become ‘more observant’ when looking at sites and pages created by others” (Braun, 2000, p. 28).

Though information literacy educators connect HTML knowledge to an advanced conceptual understanding of Web communication, HTML coding itself is often portrayed as a “low skill” (Kotamraju, 1999, p. 471). It is not a programming or scripting language after all and is considered easy to learn by comparison (Gordon, 2005; Connolly, 2012). Computer science education has argued, for example, that Web design should not be included in their curricula because “markup languages are too easy, markup languages aren’t programming languages, or…teaching web design is akin to teaching students how to use a word processor” (Rosmaita, 2006, p. 270). This attitude also likely persists because HTML was enfolded by computer literacy expectations (Carter, 2006; Kalman & Ellis, 2007) and because Web designers were/are predominantly self-taught (Deek et al., 2000; Karper, 2004; Kotamraju, 1999)—i.e., HTML can be considered easy enough for students to teach themselves. More recently however, educators' attitudes are shifting:

> It seems as if basic Web page creation with HTML is getting more complex and has too steep of a learning curve for the majority of users. People who created their own Web sites in the 1990s now hire Web designers for a site redesign. (Notess, 2006, p. 45)

The issue is not that HTML itself has become prohibitively complex—as a computer language, HTML has actually been standardized by governing bodies like the W3C, especially through the creation of XHTML and HTML5 specifications (Clark et al., 1997; Hofstetter, 2006); also, Internet browsers that initially failed to support HTML fully now do so, thus making HTML even more compatible (Wilton-Jones, 2011). Instead, the issue is that the Web design or development process now requires so much more than just learning the one computer language (Connolly, 2012). Recall Teague's (2006) summary of current expectations placed on Web designers:

> Once upon a time creating Web pages was no more difficult than using a word processor. You learned a few HTML tags, created a few graphics, and presto: Web page. Now, with streaming video, JavaScript, ASP, JSP, PHP, Shockwave, Flash, and Java, the design of Web pages may seem overwhelming to anyone who doesn't want to become a computer programmer. (p. xi)

Regardless of whether or not it is taught as a 'low skill', novice Web design students can still experience difficulty learning HTML: "If you have never designed a Web page using HTML coding, you probably find this language quite complicated with the use of the less than (<) and greater than (>) signs to begin and end all instructions or tags" (Niess et al., 2008, p. 190). The "clunky, command-driven interfaces" (Liu & Downing, 2010, p. 278) used for coding can also be alien to non-technical learners or students without prior computer language or Web programming experience (Liu & Downing, 2010). And, the practice of coding itself requires extensive accuracy, a difficulty for students that lack attention to detail (Karper, 2004). Students frequently make HTML syntax errors or typos, because they are not used to focusing on very small portions of inflexible computer language code (Park & Wiedenbeck, 2011; Voegele, 2006) and because they have difficulty seeing their own coding mistakes (Park et al., 2013, Stepp et al., 2009). In addition, Web design students may not recognize the semantic difference between display markup languages like HTML, and Web programming languages like PHP: "To the novice… such differences must be made explicit, and conveying these differences is key not only to the novice's understanding of what kind of thing HTML is, but to reducing what might be called code anxiety" (Gordon, 2005, p. 60; see also Blackwell, 2002). Learning to identify and problem-solve HTML code can require extensive and repeated practice (Karper, 2004; Maddux et al., 2008), and even then many students may not be able to code HTML successfully without the intervention of automated functionality in Web editing software, like code validators (Hofstetter, 2006; Rosmaita, 2006): "HTML and CSS validators can identify incorrect coding and point to a solution. For advanced users, these tools are an important part of any validation process" (Foley, 2002, p. 70). If Web design students are at first intimidated by learning HTML though, they can of course become more comfortable through practice (Lim et al., 2003).

Web design students also struggle with a conceptual understanding of how HTML operates as more than just markup: "HTML authoring…is not merely the tagging of text with formatting instructions, but, more generally, the creation of a hierarchically organized source document in which every element has certain possible properties" (Gordon, 2005, p. 52). For example, HTML markup tags are added to Web pages based on the meaning and priority of the content—this is what is meant by the term 'semantic Web design' (Lie & Bos, 1999). HTML must be further understood conceptually for its role in the "source-rendering relationship" (Gordon, 2005, p. 53):

> If students who are unfamiliar with how Web pages are constructed view a document containing HTML source text and then view its rendered version within a Web browser, they will see that the former bears no direct visual relationship to the latter (p. 51)…the presence of both source and rendered versions of hypertext documents adds a layer of complexity to the composing process that is typically not present in the desktop- publishing environment. (Gordon, 2005, p. 53)

(To clarify the concept of 'rendering': a Web browser is a software "necessary to translate the language with which a web page is written into an image on your screen…[it] reads HTML and then displays it as the web page you are familiar with" (Lever-Duffy et al., 2005, p. 245), i.e., browsers render HTML/CSS computer language into visual display.) Web design students are then faced with learning how to achieve visual design using not just graphic design concepts, but also computer language concepts—an incredibly advanced task for learners that may already need help "matching the concepts in their head…with the terminology for those concepts…as presented by the software and/or presented in the markup language for making Web pages" (Karper, 2004,

p. 139). Students must learn to conceptualize how the code as a textual representation translates into a visual representation in an Internet browser, and “a very large number of studies have observed that learners find translating between representations difficult” (Ainsworth, 2006, p. 189): “teaching learners to coordinate MERs [multiple representations] has also been found to be a far from trivial activity…[because] learners may find it difficult to see the relationship between such different forms of representation” (Ainsworth, 2006, p. 190). How code translates into display represents an “abstract-iconic” (Ainsworth, 2006, p. 190) level of abstraction for learners, since the HTML/CSS code does not have a concrete perceptual relationship to a Web page’s visual display. This represents yet another reason why educators advocate the use of software with WYSIWIG functionality for teaching Web design’s visual aspects (Clark et al., 1997; Descy, 1999; Dick, 2006; Hofstetter, 2006). Teaching Web design via Web editing software may be problematic, but achieving visual design via computer language represents an immeasurable abstraction (Ariga & Watanabe, 2008; Gordon, 2005; Karper, 2004; Park & Wiedenbeck, 2010)—this leads us to the lack of Web design instruction on CSS.

Cascading Style Sheets (CSS). Cascading Style Sheets (CSS) is a computer language used in conjunction with HTML/XHTML to style and control the presentation, the visual design and behavior aspects of Web pages (Andrew & Yank, 2008; Collison, 2006; Lie & Bos, 1999). As opposed to HTML, CSS was

> designed to allow precise control—outside of markup—of character spacing, text alignment, object position on the page, audio and speech output, font characteristics, etc. By separating style from markup, a web designer can simplify

> and make web contents more accessible at the same time. (Liu & Downing, 2010, p. 276)

CSS's extensive creative potential frees Web page creators from HTML's limitations as a design medium (Foley, 2002; Powell, 2010)—again, HTML was never intended to be used for layout and design (Andrew & Yank, 2008)—and CSS knowledge is now a requisite skill for Web professionals because "superior visual design affords credibility and distinctness to Web pages" (Ariga & Watanabe, 2008, p. 816):

> CSS3 is not going away. This is how we're all going to be building sites in the future. Knowing CSS3 is an increasingly important and marketable career skill. Right now, it's something that sets you apart as a top-notch designer or developer. Sooner than later, it will be something that's expected of you. (Gillenwater, 2011, p. 19)

Yet, much Web design instruction fails to thoroughly include CSS topics (Liu & Downing, 2010). (To date, there is also little academic scholarship even covertly focusing on CSS.) This again may be attributed to the subjectivity associated with teaching visual style, which CSS mediates (Ariga & Watanabe, 2008; Beriswill, 2005; Taylor et al., 2005; Victor, 2002). The overreliance on WYSIWIG software in instruction has also served as a deterrent to thorough inclusion or investigation (Maddux et al, 2008)—students may be encouraged to design their Web page's layout using the Dreamweaver interface rather than with CSS code, for example. Another likely reason for CSS's exclusion is simply limited class time. Both Karper (2004) and Taylor et al. found that "artistic aspects took longer for [students] to master" (Taylor et al., 2005, p. 339).

Jakovljevic et al. (2004) also concluded that "*the lack of sufficient time impacted on the quality of the teaching and on the learners' design solutions*" (p. 280).

However, the largest hurdle to teaching CSS—why it is not afforded the attention (or time) of other Web design instructional topics—is the conceptual and procedural complexity that CSS adds to the Web design process—"for beginners learning to design their first web sites, today's CSS can be shockingly difficult to work with. CSS is just too hard." (Andrew & Yank, 2008, p. 2). And,

> CSS does present the hypertext author with a new layer of composing problems. CSS positioning declarations, for example, are less compatible across browsers than the sort of positioning HTML can achieve through the use of tables. Also, it can take some practice to understand how the cascading principle plays out in a document's rendering, particularly when one's style sheets become lengthy and complex. (Gordon, 2005, p. 67)

As a computer language, CSS is actually "less intimidating to look at than HTML [and] CSS documents are both simpler in appearance and use a more natural-sounding vocabulary" (Gordon, 2005, p. 66). But, when learning CSS, Web design students are faced with trying to achieve visual design using not one, but two computer languages (HTML and CSS). Visual design of Web pages already represents an abstract and subjective process; composing visual design using the complexity of CSS requires even further abstraction (Ariga & Watanabe, 2008). It can take extensive time and practice for students to grasp how CSS code translates into display on a Web page, for example. As Budd, Moll, and Collison (2009) articulate it,

> most people will have been developing sites using CSS for some time before they fully grasp the intricacies of the box model, the difference between absolute and relative positioning, and how floating and clearing actually work. Once you have a firm grasp of these concepts, developing sites using CSS becomes that much easier. (p. 51)

In addition, employing CSS is procedurally more complex—it has more terminology than HTML, since its syntax contains properties with variables (like measurements) that can be used in multiple if not infinite combinations. (There are actually three kinds of style sheets, which can further complicate understanding (Hofstetter, 2006).) HTML's syntax on the other hand consists of a limited number of tags (Lie & Saarela, 1999; Teague, 2008) that are specifically governed by content.

For these reasons, even educators including CSS in their Web design instruction voice opposition to requiring it—Gordon (2005) concludes for example: "even though some CSS is very easy to implement, I do not recommend requiring novice authors to use the system" (p. 67). Other educators are more expectant though, and encourage the use of Web editing software as a 'springboard' to teaching CSS: "After looking at the preceding examples of the three kinds of cascading style sheets, you may worry that learning CSS will be difficult. Fret not, Dreamweaver, FrontPage, and NVU have style sheet editors built in" (Hofstetter, 2006, p. 285). Liu and Downing (2010) also report that their Web design "students were excited that the [CSS] exercise helped them better understand how to apply and use CSS, especially in a visual environment using no code" (p. 278) while arguing that both teachers and students can and should learn CSS. Web programming instructors are also insistent that CSS be included because that field considers it a required or even basic Web development skill: “HTML and CSS are two fundamental technologies that must be mastered by a web programmer.” (Wang & McKim, 2013, p. 69)

Regardless of opposition, “graduating from HTML-based formatting to CSS-based formatting is an important step for all hypertext authors” and CSS “should not be viewed as the exclusive province of Web development experts” (Gordon, 2005, p. 64). The importance of learning visual design aspects of Web page creation and CSS’s position as an expected Web design standard make CSS a crucial part of a Web design curriculum (Maddux et al., 2008): Web design “courses should include instruction in both verbal and visual elements, and students should be encouraged to reflect on the complex interrelations of form, content, design, and information (Wysocki, 2001)” (Turnley, 2005, p. 133).

Conceptual Learning of Web Design

As described, Web design instruction most often focuses on teaching coding and/or software, procedurally and factually. Students are taught the processes of creating Web pages, because those processes have been identified by practitioners and professionals (as evidenced in textbooks and instructional websites, if not instructors’ own experiences). Sometimes Web design lessons target concept-based learning, but often they do not, because it adds another level of difficulty for students who may already struggle to learn the software and/or coding practices (Dick, 2006). Whether for HTML, CSS, software or more abstract topics like visual design though, the practices of Web design represent extensive conceptual complexity that requires abstract thinking and problem-solving during learning (Andrew & Yank, 2008; Deek et al., 2000). An example: recall that HTML tags are added during Web page creation not just for function, but also hierarchically, semantically and visually—they must be conceptually understood by students as part of the source-rendering relationship (Gordon, 2005). A single HTML element like a hyperlink then can “embody problems of textual rhetoric (links are often short phrases), visual rhetoric (whether textual or not, links must be distinguished visually), and information architecture (links usually take the reader to a different place in a complex information network)” (Gordon, 2005, p. 54). Thus, students may be able to activate a link procedurally using HTML in two or three steps, but designing a link often requires extensive analysis and decision-making, since it functions as not only a content component but also as part of a larger conceptual/ navigational scheme (Burch, 2001). (Students may also need to design around even more abstract ideas such as user satisfaction and motivation (Zhang & von Dran, 2000).) When creating links, students must ultimately understand that

> the web…is not simply a tool for accessing and retrieving paper documents in a digital form. Instead, it is much more. It is also a participatory technology that requires an understanding of the production and distribution of text and image files through an expansive public network. (Mackey & Ho, 2005, p. 543)

Students can struggle immensely to comprehend and execute just this one Web design concept, “because creating hyperlinks for Web pages is fundamentally different from any aspect of traditional print-based writing…the concept proved to be quite overwhelming for the students who were unfamiliar with the process of composing Web pages” (Dick, 2006, p. 211).

Teaching for conceptual understanding. Instructors are then faced with teaching not just facts and procedural knowledge, but also reasoning and understanding (Ariga & Watanabe, 2008) to Web design students who are most often "procedural rather than conceptual learners" (Karper, 2004, p. 139; see also Jakovljevic et al., 2004). Karper (2004) found that her

> beginning designers relied heavily on creating and memorizing procedures for working with the technologies of Web design. This type of learning is called procedural learning, which is distinct from conceptual learning, the more flexible acquisition and application of interrelated concepts in order to accomplish a task. (p. 52)

Because of their prior experience and comfort with procedural instruction, Karper's (2004) students "didn't care so much about abstract concepts such as 'Web-safe color' or 'table-based layout'" (p. 167) and instead

> attempted to reduce all technological aspects of Web design of the Web page process to a series of precise context-specific steps which they could then repeat over and over again, rather than approaching technology use in this situation as a series of flexible concepts and procedures that could be transferred to different situations. (Karper, 2004, p. 133)

This finding suggests that teaching Web design as a procedural process is necessary—"multimedia and hypertext design must be taught as a process, similar to the ways in which rhetoric and composition approaches and teaches 'traditional' writing" (p. 22), Karper (2004) concludes—but it should also strive for intellectual educational goals, regardless of students' inexperience or discomfort as conceptual learners. For example,

> it is easy for most of the time and emphasis on the class to become focused on teaching markup and helping students solve problems in their syntax…in order to avoid neglect of educational issues, students need to constantly be encouraged to think about learning and teaching variables. (Maddux et al., 2008, p. 4229)

In other words, Web design educators now recommend that instruction should evolve to be concept-based and to include the "explicit teaching of thinking skills" (Jakovljevic et al., 2004). Otherwise, students' overreliance on procedural learning can be detrimental to their ability to problem solve and transfer knowledge when practicing Web design: "technological problem solving is open-ended and creative depending on the problem" and relying on "a systematic, step-by-step guided process" is often not effective, even though it may be what students demand (Jakovljevic et al., 2004, p. 285). This is in line with the call by education scholars in general to promote learning for understanding, rather than just content coverage:

> More than ever, the sheer magnitude of human knowledge renders its coverage by education an impossibility; rather, the goal of education is better conceived as helping students develop the intellectual tools and learning strategies needed to acquire the knowledge that allows people to think productively... (Bransford et al., 2000, p. 5)

Web design students should not simply 'borrow' the procedures or opinions of textbooks or experts (Wiggins & McTighe, 2005), but instead must be guided towards "deep understanding of subject matter [that] transforms factual information into usable knowledge" (Bransford et al., 2000, p. 16). This "focus on the necessary critical and intellectual approaches [is] missing from many Web design curricula" (Royal, 2005, p.

401) and scholars such as Royal (2005) now argue that "teaching without a focus on integration, judgment, and perspective in the new media environment [i.e., the Web] will ultimately fail students who suddenly find their skills outdated, outmoded, or out of sync with the real world" (p. 412), particularly as software, computer languages and even communication trends evolve.

Competency as a conceptual learner is then an important characteristic for Web design students; the capability to experiment and invent in the form of design, and the confidence to take risks, is essential (Gordon, 2005; Karper, 2004). Deek et al. (2000) found in their analysis of successful Web developers for example, that Web page creators needed the following characteristics:

- Willingness to risk
- Awareness of their own strengths and weaknesses
- Knowledge of context and culture
- Ability to negotiate with multiple influences
- Ability to articulate a problem
- Ability to take action and test barriers (pp. 44-48)

Again, these are largely conceptual capabilities, rather than procedural or even technological. Higher-education students also need experience as conceptual learners, because academically,

> today's digital divide may become less concerned with economic conditions and access to technology and more concerned with an individual's technical and conceptual competence. A new digital divide has emerged out of the students' abilities to naturally adapt new technological advancements and the instructors'

> inability to adjust their instructional modes and strategies to meet the ever-changing capabilities of increasingly sophisticated and technologically savvy students. (Kalman & Ellis, 2007, pp. 37-38)

If nothing else, conceptual learning capabilities are necessitated by the rapidity with which the Web and Web design practice evolve (Kotamraju, 1999)—"Web site developers have to adapt to and adopt new technologies, they have to be able to learn new techniques by themselves, from other developers, or other sites to create successful web pages and to constantly upgrade their sites" (Deek et al., 2000, p. 42; see also Wang & McKim, 2013). Recall that in a Web design class, "the worlds of the academic and professional are intersecting" (Walker, 2002, p. 66), and as such, concept-based Web design instruction holds professional importance—even the professional emphasis is on the conceptual:

> Effective professional practice requires more than knowing what tools and techniques are available and how to use them. The hallmark of professional practice is the ability to select and use tools and techniques to devise a solution that meets the demands of a particular situation. This requires the flexibility and adaptability that come from understanding at the level of theoretical principle rather than at the more superficial level of technique. (Newby, Stepich, Lehman, & Russell, 2006, p. 26)

Newby et al. (2006) here address general educational technology standards, but their comments are in line with the evolving definitions of information/computer literacy that incorporate Web design (Mackey & Ho, 2005):

> Today, the emphasis has shifted away from learning software features, which we believe students can learn independently, and more toward using computer tools to accomplish real-world tasks. Learners must be able to integrate their knowledge of technology skills, visual literacy, analytical skills, and critical thinking skills to solve complex problems. (Kalman & Ellis, 2007, p. 24)

This is a continuation of the literacy argument that "technical proficiency by itself…does not allow students to become successful web developers. To foster student rhetorical agency, technical instruction must be integrated with rhetorical analysis, medium-specific concerns, and considerations of larger cultural contexts" (Turnley, 2005, p. 132).

Instructional Strategies

How then to bridge the gap between students who are procedural learners and the call to teach Web design conceptually? Kalman and Ellis (2007) find that these goals and strategies should be included:

> a. Individualize the course to better accommodate students' broad range of entry skills,
>
> b. engage students in meaningful assignments,
>
> c. promote student control and active learning,
>
> d. encourage the use of alternative learning strategies (i.e., self-instructional tutorials), and
>
> e. transform the instructor's role from a software demonstrator to a facilitator of inquiry. (pp. 27-28)

Students that lack conceptual learning capability can be eased into the uniqueness of Web design with "direct, concrete kinds of experiences" (Smaldino et al., 2005, p. 49), like procedural tasks and engaging with multimedia on the Internet, or with experiential learning that

> goes beyond that of simple ‘‘problem based learning’’ (e.g., Moesby, 2002). This approach can be illustrated by [a] discursive approach taken to discovering HTML. Firstly, students were presented with the basic tools to create a web page in HTML such as a text editor, and a browser. Next, a description of definitive boundaries such as knowledge of the different kinds of tags, and how to define tables and paragraphs was presented. It was from these two abstracts that the student could then start constructing their knowledge base. (Taylor et al., 2007, p. 223)

Royal (2005) also recommends that Web design instructors bridge the gap by modeling their own conceptual (and procedural) understanding: “the instructor must illustrate to students the processes by which he/she accomplishes troubleshooting, seeks out additional information, or participates in self-study to advance knowledge of the field” (p. 411).

It should be noted that some conceptual learning tasks, like analysis, have been found insufficient to teach Web design (Marx, 2003). Karper (2004) concluded during her study that “the current analysis-based, technocentric methods of teaching design are not enough to help designers learn to produce ‘good’ Web pages” (p. 20-21); “Critique and analysis are important skills and tools, and should not be neglected or discarded in scholarship, learning, or teaching…[but] “analysis-based teaching is not leading to pages that meet the standards given for and generated by analysis…ability to analyze does not always imply ability to produce” (Karper, 2004, p. 4). This finding contradicts the "dominant model" for teaching Web design, which involves "three key components: analysis of existing Web pages, acquisition of technological skills for Web page production, and collaborative work in creating design" (Karper, 2004, p. 12). The finding also contradicts arguments that in Web design instruction "the focus should be on what you want to achieve, not on how to do it" (p. 17) and "the primary reason to use the Web design approach is to help students learn content" (Lim et al., 2003, p. 18). This also casts Web design's use as a constructivist activity in the service of learning separate, non-technical topics in a problematic light. In order for Web design instruction to be effective in a teaching with technology assignment for example, educators must not avoid teaching the *practice* of Web design.

The argument then becomes that "a balanced combination of theory and practice is essential in studying web design. Only those who manage to combine these two things can become good web designers" (Krunić et al., 2006, p. 326); "it is necessary to teach technology in contexts that honor the rich connections between technology, the subject-matter (content) and the means of teaching it (the pedagogy)" (Koehler & Mishra, 2005, p. 148). Again, technical Web design knowledge is essential, but should be taught in regard to the design or communication concepts which necessitate it (Gordon, 2005; Royal, 2005). Karper (2004) provides an example of an effective conceptual learning scenario:

> *[C]onceptual learning* is used…to talk about the ways in which more experienced Web designers may understand and use technologies by abstracting sets of procedures into concepts, or understanding processes of Web design as sets of concepts which encompass multiple possible procedures that they can choose as

> appropriate rather than working through one procedure without modification…this richer conceptual understanding may contribute to a richer and more refined composing process. (Karper, 2004, p. 53)

It is, after all, one of Karper's (2004) central conclusions that the "composing processes for Web designers are substantially different from composing processes in other media, a finding which has serious implications for research and teaching" (p. 107). For example, students may begin by imitating the content and design of existing Web pages, but they quickly discover that their existing composing strategies do not transfer and they must develop new conceptual ones through experimentation. Beriswill (2005) found further evidence of this:

> In order to explore various attribute options for graphics and page layouts, designers would branch out of their ongoing design process and enter an experimentation stage where they played with and explored a particular attribute until they found a combination that worked together visually. (p. 921)

Karper (2004) labels this invention and experimentation process as remediation—"remediation is different from imitation in that the process of remediation transforms and repurposes existing content to fit the perceived needs of the new medium" (Karper, 2004, p. 50)—and goes on to suggest that "we require a new model of the [Web] composing process which accounts for the use of identification and remediation practices in the planning stage as well as the greater role that technologies play in mediating different aspects of the process" (p. 144). This idea of a conceptual Web design model is picked up by other educators (e.g., Hazzan, 2004; Jakovljevic et al., 2004), but like Web design scholarship in general, there is little consensus. The relevant point is that these models promote learning Web design as a conceptual process, not just procedural or factual.

Modern learning theories. In order for Web design instruction to effectively shift towards conceptual learning, it must also incorporate the teaching strategies endorsed by modern learning theories: “To be most effective, LWD [Learning Web Design] should be supported by sound instructional strategies. Motivational strategies, feedback mechanisms, and instructor expectations should be carefully planned and reviewed” (Lim, Plucker, & Bichelmeyer, 2003, p. 18). Of course, “there is no universal best teaching practice” (Bransford et al., 2000, p. 22), but education has identified successful teaching methods like didactic (direct) instruction, constructivist facilitation, and performance coaching, and “to teach for understanding requires the routine use of all three types” (Wiggins & McTighe, 2005, p. 242). These methods in turn are informed by learning theories such as behaviorism, constructivism, and information processing (cognitivism), and

> principles from the different theories can be applied to virtually any learning situation… [For example,] reinforcement (from the behavioral perspective), organized information (from the information processing perspective), and learning from one another (from the constructivist perspective) are principles that will be useful in virtually every instructional situation. (Newby et al., 2006, p. 39)

In behaviorist theory, learning is “a response to external stimuli…the learner acquires behaviors, skills, and knowledge in response to the rewards, punishments, or withheld responses associated with them” (Lever-Duffy et al., 2005, p. 14).). It often focuses on direct methods of instruction (e.g., lecture or demonstration), where “the goal of learning is to efficiently transmit knowledge from the instructor to the learners…Students are passive recipients of knowledge, rather than constructing their own knowledge" (Hadjerrouit, 2005, p. 118). Information processing theory or cognitivism focuses instead on learning as a

> mental operation that takes place when information enters through the senses, undergoes mental manipulation, is stored, and is finally used…this theory makes mental activity (cognition) the primary source of study. Although behavior is still considered critical, it is viewed as an indicator of cognitive processes rather than just an outcome of a stimulus-response cycle. (Lever-Duffy et al., 2005, p. 15)

Information processing emphasizes the "critical role memory plays in helping [learners] translate new information into a form they can remember and use" (Newby et al., 2006, p. 31) and targets instructional methods like discussion, schema building, or even multimedia use that help students organize new information and link it to existing information (Newby et al., 2006). It also extensively investigates and strategizes how to avoid overloading students' limited cognitive processing capabilities (DeLeeuw & Mayer, 2008; Schnotz & Kurschner, 2007).

Finally, in constructivist theory, "knowledge is constructed as learners try to make sense of their experiences. Learning, then, is a continuous process of experience and reflection in which learners create, test, and refine mental models that will synthesize their experience" (Newby et al., 2006, p. 35). Learning is seen as a mental operation, like within information processing, but "in the information processing perspective…the assumption is that knowledge is objective and can be described separate from the knower…The assumption in the constructivist perspective is that knowledge cannot be separated from the knower" (Newby et al., 2006, p. 34). Constructivism embraces instructional practices like the solving of authentic problems, collaboration or teamwork, discovery learning, and self-evaluation in its promotion of student autonomy and initiative (Hadjerrouit, 2005; Lever-Duffy et al., 2005; Smaldino et al., 2005): "constructivism is more demanding in terms of communication with students and provision of intrinsically motivating learning activities that are situated in real-world environments" (Hadjerrouit, 2005, p. 129). Scaffolding of students' learning by the instructor is also a crucial strategy within constructivism:

> A scaffold is a support that a teacher or learning environment provides to a learner to assist him or her in a range of cognitive tasks, from the understanding of a task and mastering of a skill to the solving of a problem. Scaffolding is an important feature of Vygotsky's (1962) social development theory… this theory holds [that learning] requires social interaction through expert guidance and peer collaboration. (Doering & Veletsianos, 2007, p. 109)

These learning theories are not in themselves instructional strategies, and as to be expected, the instructional approaches taken must depend on each Web design situation's specific goals:

> Some theories fit some learning situations better than others…this fit depends on two critical factors: students' knowledge level and the amount of thought and reflection required by the learning tasks...students with little content knowledge are likely to benefit most from learning strategies based on the behavioral perspective. As students' knowledge grows, the emphasis may shift to the

information processing perspective and then the constructivist perspective. (Newby et al., 2006, p. 39)

This coordinated transition through the theories' methods and practices has been termed the "new science of learning" (Bransford et al., 2000, p. 3); it is a unified theory that endorses active learning strategies and promotes teaching for understanding and transfer. It also endorses instruction that is not just learner-centered and knowledge-centered, but also assessment-centered, and community-centered.

The New Science of Learning:

- The goal of education is better conceived as helping students develop the intellectual tools and learning strategies needed to acquire the knowledge that allows people to think productively (p. 5)
- Emphasis should be placed on learning with understanding (p. 8)
- Facts are still important for thinking and problem solving (p. 9)
- Focus should be on the processes of knowing (p. 10)
- People construct new knowledge and understandings based on what they already know and believe (p. 10)
- Emphasis should be placed on the importance of helping people take control of their own learning (p. 12)
- Students should be better prepared to transfer what they have learned to new problems and settings. (p. 13)

(Bransford et al., 2000)

Figure 3. Summary of the new science/theory of learning.

Web design educators and scholars like Jakovljevic et al. (2004), Karper (2004), Royal (2005), and Kalman and Ellis (2007) have been tacitly mirroring the tenets of the new science of learning by arguing that Web design be taught conceptually, with an emphasis on self-instructional techniques.

The needs of novices. Because Web design students lack both content knowledge and prior experience from which to draw upon when creating Web pages, they have specific learning needs that must be strategized for: "for students to overcome their lack of confidence, insufficient knowledge of task requirements, and difficulty applying strategies, instruction in conspicuous strategies and careful scaffolding of the process need to be implemented" (Rockwell, 2008, p. 110). Bransford et al. (2000) conclude that "it would be a mistake simply to expose novices to expert models and assume that the novices will learn effectively; what they will learn depends on how much they know already" (p. 50). In fact, "most expert big ideas are abstract and *counterintuitive* to the novice, prone to misunderstanding" (Wiggins & McTighe, 2005, p. 67). This means that Web design instructors need to adopt strategies that may seem counterintuitive: "Teachers, especially at the high school and college level, often fail to adequately consider the deficiencies in the students' prior *experiences*—and then wrongly think that what they need is more *knowledge*" (Wiggins & McTighe, 2005, p. 208). What novices actually require is experience and engagement: "Gagné's research revealed that well-designed lessons begin with the arousal of students' interest and then move on to present new material" (Smaldino et al., 2005, p. 49). Novices can also easily be overwhelmed by large amounts of new information, which hampers their eventual understanding:

> Attempts to cover too many topics too quickly may hinder learning and subsequent transfer because students (a) learn only isolated sets of facts that are not organized and connected or (b) are introduced to organizing principles that

> they cannot grasp because they lack enough specific knowledge to make them meaningful. (Bransford et al., 2000, p. 58)

Novices learning to code and program are particularly susceptible to this cognitive overload: “It has been observed that a large number of students achieve only low grades and become disillusioned with the subject [computer programming]…one of the reasons for the above is that students experience very high cognitive load during their learning” (Garner, 2002, p. 578). Students instead “need to take time to explore underlying concepts and to generate connections to other information they possess” (Bransford et al., 2000, p. 58), before they can effectively absorb new facts or knowledge; “new knowledge is either assimilated (fitted into existing maps) or accommodated (existing maps are adjusted to accommodate new information)” (Lever-Duffy et al., 2005, p. 15). Learning is enhanced when instructors “pay attention to the knowledge and beliefs that learners bring to a learning task, use this knowledge as a starting point for new instruction, and monitor students’ changing conceptions as instruction proceeds” (Bransford et al., 2000, p. 11). (Called “learner-centered instruction,” this teaching method creates “environments that pay careful attention to the knowledge, skills, attitudes, and beliefs that learners bring to the educational setting” (Bransford et al., 2000, p. 133) and then matches instruction to the learners’ experiences, rather than the other way around.) An example: Web design novices may lack even ‘remedial’ computer or Internet literacy experience, for which instruction must compensate:

> Although mastery of basic literacy skills is becoming less of a problem with each passing semester, we still find that some students come to us without the ability to use a Web browser efficiently or to use a search engine to quickly and easily find

> information they need for their Web pages. Many more students lack any knowledge of basic home page design principles or skills needed to upload and download files between a personal computer and the campus web server. (Maddux et al., 2008, p. 4228)

If students have no prior knowledge of a subject (like in Web design), then a strategy such as "progressive formalization" may be useful. Progressive formalization

> begins with the informal ideas that students bring to school and gradually helps them see how these ideas can be transformed and formalized. Instructional units encourage students to build on their informal ideas in a gradual but structured manner so that they acquire the concepts and procedures of a discipline. (Bransford et al., 2000, p. 137)

Students' everyday experiences using the Internet and Web pages can be deconstructed to reveal how Web pages actually operate, how the display and functionality they encounter had to be purposefully designed and then built, for example. Starting with an exploration of Web pages or the Internet may still be problematic for novices though: Turnley (2005) reports that students' experiences with Web pages "tend to highlight visits to technically sophisticated, commercial sites…students often express that they were not able to find less sophisticated examples in their searches" (p. 134). It becomes difficult for them to relate or connect the rudimentary Web design ideas they are first taught with the advanced Web pages they see daily. (Karper's (2004) findings that analysis assignments do not lead students to create effective Web pages support this conclusion.) Exploring Web pages that they cannot imitate ultimately does not help novices connect Web design knowledge to their prior experience: "When producing web documents for the first time, writers easily can become intimidated, especially if they see an elaborate site as the expected culmination of their efforts" (Turnley, 2005, p. 134). Rather than starting with Web pages or technical information, it may be better to 'ease' students into Web page creation by "encouraging them to take time and focus on the planning and research stages of web design" (Turnley, 2005, p. 144), particularly if this is a process with which they are more experienced.

Regardless of technique, novices' initial experiences must be direct and concrete, because "students approaching a subject new to them learn best from structured presentations even if they have a learning style that would otherwise indicate more open-ended, unstructured methods" (Smaldino et al., 2005, p. 50). Web design novices for example need structure, because they have a "smaller repertoire of strategies and are often inventing strategies and schemas [i.e., procedures] as they acquire new knowledge" (Karper, 2004, p. 56). Their choices and conclusions do not mirror those of experts:

> A pronounced difference between experts and novices is that experts' command of concepts shapes their understanding of new information: it allows them to see patterns, relationships, or discrepancies that are not apparent to novices. They do not necessarily have better overall memories than other people. But their conceptual understanding allows them to extract a level of meaning from information that is not apparent to novices, and this helps them select and remember relevant information. Experts are also able to fluently access relevant knowledge because their understanding of subject matter allows them to quickly identify what is relevant. Hence, their attention is not overtaxed by complex events. (Bransford et al., 2000, pp. 16-17)

Instruction must then be designed around novices' inefficient organization and memorization of new information. They must be taught to recognize the patterns, chunks, or "big ideas" necessary for conceptual understanding, as well as how to remember them (Bransford et al., 2000; Wiggins & McTighe, 2005). The goal is always to avoid presenting "too many disconnected facts in too short a time—the 'mile wide, inch deep' problem" (Bransford et al., 2000, p. 24) and instead employ "instructional sequences that immerse learners early on in intriguing issues, problems, situations, or other experiences and postpone the teaching of definitions, rules, and theories until they are needed to make sense of experience" (Wiggins & McTighe, 2005, p. 220). But, memorization strategies are still needed throughout. There are a plenitude of memorization strategies, but the new science of learning targets these:

- Rehearsal (repeating items over and over), which tends to improve rote recall (Belmont and Butterfield, 1971);
- elaboration (Reder and Anderson, 1980), which improves retention of more meaningful units such as sentences;
- summarization (Brown and Day, 1984), which increases retention and comprehension;
- clustering: organizing disparate pieces of information into meaningful units. (Bransford et al., 2000, p. 96)

Even with recognition and retrieval strategies in place, novices' misconceptions and misunderstandings are to be expected:

> Because learning involves transfer from previous experiences, one's existing knowledge can also make it difficult to learn new information. Sometimes new

> information will seem incomprehensible to students, but this feeling of confusion can at least let them identify the existence of a problem (see, e.g., Bransford and Johnson, 1972; Dooling and Lachman, 1971). A more problematic situation occurs when people construct a coherent (for them) representation of information while deeply misunderstanding the new information. Under these conditions, the learner doesn't realize that he or she is failing to understand. (Bransford et al., 2000, p. 70)

Teaching strategies such as interactive demonstrations, which include worked-through examples or correct versus incorrect examples, have been found effective for avoiding misunderstandings (Bransford et al., 2000)—these also offer the structure novices require for new information. Schnotz and Kurschner (2007) concluded that "novices clearly benefited most from worked out examples with complex tasks. However, when learners became more experienced, the advantage of the worked-out examples disappeared and the exploratory group performed better than the worked examples group" (p. 473). Meaning, novices struggle with traditional problem solving tasks because, again, they do not yet have the strategies and conceptual understanding necessary for confidently exploring a problem (Schnotz & Kurschner, 2007).

Struggling with new problems or even contradictory information can be a useful learning technique for novices—"Cognitivists propose that learners build up and enrich their mental schemata when their minds are actively engaged in struggling to remember or apply some new concept or principle" (Smaldino et al., 2005, p. 67)—but the struggle must not overwhelm novices any more than the quantity of new information should. Bridging strategies, which aim to "bridge from students' correct beliefs (called anchoring conceptions) to their misconceptions through a series of intermediate analogous situations" (Bransford et al., 2000, p. 177) have also been found effective against misconceptions. As expected in learner-centered instruction, Web design teachers must continuously attempt to uncover novices' misunderstandings and guide them towards intended conclusions. This may seem commonsensical, but "numerous research experiments demonstrate the persistence of preexisting understandings among older students even after a new model has been taught that contradicts the naïve understanding" (Bransford et al., 2000, pp. 15). In learning, "the tendency is to use the same strategy or approach used the last time, thus diminishing the cognitive challenge of the problem" (Raths, 2002, p. 236-237), even if the approach is incorrect or reinforces a misconception. Instructors must then take steps to 'break' novices' from their faulty connections to prior experience. For example, if Web design students' prior experiences with HTML predate semantic coding standards (introduced around 1997, when HTML 3.0 was standardized ("HTML," Wikipedia, 2013).), or try to compare Web design to word processing or print design, instructors must uncover this and design a learning experience to counteract the misconceptions.

Motivation strategies. Strategies that address students' motivations during learning must also be considered: "motivation affects the amount of time that people are willing to devote to learning" (Bransford et al., 2000, p. 60) and learners' interest must first be stimulated and then sustained throughout instruction in order for it to be effective (Keller, 2010). Web design instruction benefits from perceptions of usefulness and marketability—recall that creating Web pages is seen as a valued professional activity by students, as well as an engaging form of authentic experience and active learning by instructors (Karper, 2004; Marx, 2003; Lim et al., 2003; Mull, 2001). But, initial engagement based on professional desirability or the novelty of the Web medium will not last (Fisher, 2001). To maintain learners' motivation, the ARCS motivational model (Keller, 2010) posits that four conditions must be strategized for: attention, relevance, confidence and satisfaction.

Attention-getting strategies are needed for capturing learners' interest and stimulating their curiosity to learn—this can include simple orienting activities or exposing them to new experiences, but to be most effective, instruction should "[create] a problem situation which can only be resolved by knowledge-seeking behavior" (Keller, 2010, p. 47). It should be accompanied by variation and change (in the activity, the classroom environment, or in the instructional materials medium, e.g.) so that learners are motivated to re-focus. Relevance-building strategies aim to reveal the "personal meaningfulness of the material" and a "successful instructor is able to build bridges between the subject matter and the learner's needs, wants and desires" (Keller, 2010, p. 48): "learners of all ages are more motivated when they can see the usefulness of what they are learning and when they can use that information to do something that has an impact on others" (Bransford et al., 2000, p. 61). The most significant motivational strategy for establishing relevance is of course the use of authentic assignments and examples:

> Authentic tasks are those based on the world of experience and work. An authentic task is a realistic, intrinsically motivating problem that is situated in some meaningful real-world environments in order to get students actively involved in the learning process. (Hadjerrouit, 2005, p. 122)

(This aligns with constructivism's call for authentic assignments as part of student-controlled or learner-centered education (Bransford et al., 2000).) Relevance-affirming activities should also motivate students to work towards a goal, whether it be professional or personal, and to be most effective should connect to learners' prior experiences and knowledge. In an echo of instructional strategies for meeting novices' learning needs, Keller (2010) even argues that "the use of concrete examples from settings familiar to the learner can help to achieve relevance, especially when teaching abstract material" (p. 50).

Confidence during learning is also important for maintaining student motivation—"it is important to provide success experiences for learners as soon as possible" (Keller, 2010, p. 50); this can be done by clearly defining expectations for success and then allowing students a sense of personal control over their success. Success is especially important for students who lack experience or knowledge (like novices)—"for such students, evidence of immediate success can be a strong motivating force for further learning" (Smaldino et al., 2005, p. 68). Strategies for confidence-building can include providing examples to work towards—again, students need a goal to motivate them—providing feedback that "supports effort and ability as the causes of success" (Keller, 2010, p. 52), rather than factors outside the student's control and, especially, providing "corrective feedback that allows them to see the causes of their mistakes and how to take corrective action" (Keller, 2010, p. 52). (It is this idea around which this study's curriculum unit's assessment is built.) Confidence issues are especially problematic for non-technical learners during Web design instruction, because they lack experience with computer languages and Web software: "computer anxiety and attitudes toward computers are two [issues] that significantly influence computer learning performance" (Chou & Wang, 1999, p. 328). Web design students may feel "code anxiety" (Gordon, 2005, p. 60) if they are fearful about making mistakes in the code, or they may struggle to know what to do next (Kalman & Ellis, 2007; Karper, 2004), because HTML and CSS are so conceptually and procedurally complex. This lack of confidence can hinder their conceptual learning as well as their motivation. For example, Karper (2004) found that

> the more confident designers took greater risks with the technology, understanding that they could always 'undo' later…I would argue that greater ease with the technological freed the designers to focus more attention and energy on the rhetorical [conceptual] aspects of their webpages…" (p. 140)

To build students' confidence, instruction should again try to motivate them by providing "experiential learning activities and other methods that require the learner to do problem solving [in] situations in which the learner has to exercise personal control to succeed" (Keller, 2010, p. 52), i.e., novices and non-technical learners can become more confident through practice and experience-building strategies (Lim et al., 2003).

A sense of satisfaction must also be maintained if students are to continue to be motivated to learn. The simplest strategy for establishing satisfaction is to establish success: "for a student to be able to successfully perform a challenging task at the end of the class that he or she could not do at the beginning is a very satisfying experience" (Keller, 2010, p. 53). Strategies for building satisfaction into instruction also include the use of meaningful, authentic experiences—this aligns with the need for relevance—case studies, and simulations (Keller, 2010). Or, some form of extrinsic or symbolic reward may be necessary to sustain a learner's satisfaction if praise or a sense of accomplishment are insufficient (Keller, 2010). When the subject matter does not easily lend itself to immediate success because it is a complex or multistep process—as is the case with Web design—then positive consequences must at least be established frequently throughout the instruction (Keller, 2010). Defining success as a student's 'personal best', for example, may alleviate concerns of not being able to meet external or professional standards. (The study's curriculum unit gauges students' progress in this way, so that individualization of assignments can occur and so that novices will not be intimidated if their Web pages do not imitate the complex Web sites they use for everyday activities (Turnley, 2005).) Also, if students can "have some feeling of control over their situation and…see the various pieces fitting into a whole" (Keller, 2010, p. 54) their satisfaction will be retained. Again, a sense of control is important for students' confidence during learning as well as satisfaction—this further exemplifies how in the ARCS model the motivational strategies function together. The ARCS motivational strategies also align with instructional strategies for meeting novices' learning needs: "designing assignments that build on prior knowledge and already attained skills facilitate both learning and motivation" (Rockwell, 2008, p. 112), for example. Multiple Web design scholars have concluded that students can find Web design to be a satisfying and motivating topic (Kalman & Ellis, 2007; Reber, 2005):

> research suggests that in a student-as-hypermedia designer approach, students are highly satisfied with the activities, develop skills and knowledge effectively, are mentally engaged to a much greater extent by developing materials than by studying materials, are highly motivated by the activity because they gain a sense of ownership in the product and in their learning… (Lim et al., 2003, p. 14)

Sustaining novices' motivation throughout learning Web design still represents significant difficulty—as mentioned, the novelty that captures their attention fades as the procedural and conceptual complexity unfurls (Fisher, 2001). The challenge that Web design represents "must be at the proper level of difficulty in order to be and to remain motivating: tasks that are too easy become boring; tasks that are too difficult cause frustration" (Bransford et al., 2000, p. 60; see also Garner, 2002). The use of conceptual learning strategies in Web design instruction, which can represent more difficulty for novices than procedural learning (Jakovljevic et al., 2004; Karper, 2004), must then be carefully considered for its impact on students' motivation to continue learning Web design.

Instructional Materials

The shift towards Web design instruction that promotes conceptual learning has implications for the materials and resources used. As mentioned prior, "there are many how-to manuals for building Web pages in bookstores and on websites" (Ariga & Watanabe, 2008, p. 815). Most of these textbooks and websites focus on procedural learning though, employing step-by-step tutorials to cover the content and processes needed to create Web pages. The textbooks or websites may pay only minimal attention to visual design or other conceptual topics (Ariga & Watanabe, 2008; Clark et al., 1997); this can limit Web design students' learning in the same manner that Web-editing software constrains their experience. Wiggins and McTighe (2005) summarize the argument:

> Textbooks distort how understanding develops, in the expert and the novice, by presenting only the cleaned-up residue. You simply cannot learn to 'do' the

> subject or understand it in depth by studying only a simplified summation of findings…Few textbooks are designed around a series of defining experiences, yet well-designed experience is the only way to make ideas real. (pp. 232-233)

Web design's instructional materials may be designed around practice, if not experience-building, but their reliance on procedural learning skills still fails to advance students' conceptual learning experience. Wiggins & McTighe go on to contend that "in depth teaching for understanding using a problem-based approach supported by small texts provides far better results than the typical overloaded textbook focused U.S. approach" (2005, p. 312): "Much of what we call expert knowledge is the result of trial and error, inquiry, and argument…when we teach only from textbooks….students are easily mislead into believing that knowledge is somehow just there for the picking" (Wiggins & McTighe, 2005, p. 235).

Another common criticism aimed at Web design's instructional materials is that they present information in a techno-centric manner that only makes sense to technology experts (Robson & Freeman, 2012). The resources suffer from the authors' "Expert Blind Spot" and do not meet the learning needs of novices or non-technical learners (Bransford et al., 2000; Wiggins & McTighe, 2005): "From the perspective of the expert, jargon and shorthand phrases permit easy and efficient communication; to the novice they are often off-putting barriers to understanding" (Wiggins & McTighe, 2005, p. 139). In line with this argument, Karper (2004) reports that "because beginning writers/designers are deriving more of their strategies during creation, and because they have fewer strategies, they often make choices that experienced designers would not" (p. 56). Web design instructors "need to know where first-time designers may encounter problems or need support that is different from what is currently being provided" (Karper, 2004, p. 26-27). Being techno-centric, the instructional materials may even fail to use simple tactics like graphics, and as a result, non-technical learners may not find them useful: "It may sound like a trivial aspect of higher education, but the visual appeal and usability of the information being taught does affect how willing a student is to use the material" (Burch, 2001, p. 362). Karper (2004) for example found that her novice Web design students rejected how-to "documentation because it didn't allow them to match the concepts in their head with how to do it with the technology" (p. 140); "the fact that the novice designers didn't use documentation and didn't find it helpful is an important piece of information, albeit not a surprising one" (Karper, 2004, p. 170). Her study participants relied heavily on lecture notes and asking the instructor (the expert), because they could use their own terminology:

> Participants often focused on their need for direct instruction from others in order to help them learn, or their need to have a resource person available in case they get stuck. Only two participants made reference from learning from documentation or printed instruction …the others emphasized that they needed to work with an actual human being in order to learn best. This common preference for learning by working with another person may explain some of the designers' reluctance to use documentation or online help to help them with their Web design tasks, particularly when their preferred method of learning support (an actual person) was available most of the time. (Karper, 2004, p. 85)

Maddux et al. (2008) also conclude that lecture notes function effectively in place of a textbook, but they still advocate one, arguing that it should be "used as a reference book rather than as a traditional textbook" (p. 4231)—this is the tact for using instructional materials advocated by Wiggins and McTighe (2005), as well as other instructional design experts.

Web design students *do* still need extensive reference materials, and the skill to interact with them is an essential conceptual learning capability. For example, novice Web designers often need multiple models—reusable, re-creatable examples that can be used for identification and remediation, e.g. (Karper, 2004)—that enable them to develop conceptual learning skills, like "the ability to recognize similarities and differences between the problem to be solved and previous problems and their solutions" (Hadjerrouit, 2005, p. 124). Mull (2001) also recounts that when his students could effectively interact with reference materials, they "felt more secure, as did I, because, when doing assignments on their own, they did not use me as a crutch to answer questions for them" (p. 29). Kalman and Ellis (2007) also report that their students were willing to "make greater use of self-instructional learning resources such as print guides, videotapes, [and] online tutorials" (p. 30).

The identification of the materials as 'self-instructional' is common among Web design resources, and, for Web design students in particular, development of self-instruction study skills is essential. The dual issue for instructors is that non-technical students now need to learn how to self-instruct, in addition to learning Web design's content and concepts using the technology-centric self-instructional materials available (Liu & Downing, 2010). And, just as most Web design instructional materials do not strive to engage students' conceptual learning skills, they also do not advance their study skills:

> Far too often it is *assumed* that students will somehow already possess key enabling skills (e.g., study skills, public speaking skills, graphic design skills, group management skills)—with the unfortunate results that cause more educators to complain about the absence of those skills than to target them in their planning. Helping students to "learn how to learn" and "how to perform" is both a vital mission and a commonly overlooked one. (Wiggins & McTighe, 2005, p. 59)

As evidenced by Karper's (2004) findings alone, Web design educators cannot assume that non-technical learners will be able to self-instruct, or that the instructional materials will be meet students' conceptual learning needs.

Web-based instructional resources. To compensate for textbooks' limitations, some Web design educators argue that "the most effective teaching resources are the examples presented from different websites and our own daily use of technology" (Royal, 2005, p. 412), e.g., using Web pages. The use of Web-based instructional materials is often promoted as an evolution from print materials, or even a remedy to them, but, as Lim et al. (2003) are quick to point out, "the web is used mostly as 'the latest manifestation of a textbook' in college classrooms" (p. 14)—the Web is still "predominantly a text based medium, best suited to verbalisers" (Parkinson & Redmond, 2002, p. 42), and so instructors simply employ Web-based materials as they had printed pages. It should still be recognized that Web-based materials are increasingly seen as viable alternatives to textbooks (Bransford et al., 2000; Kahn, 2011; Wiggins & McTighe, 2005). Kalman and Ellis (2007) describe this scenario for their computer literacy course:

> We eliminated the required textbook because the students appeared to prefer using web resources and because once the technical skills were mastered, the textbook had little, if any, future value for the students. We provided a broader variety of learning resources such as videotapes, print-based tutorials with CD-ROMs, web-based tutorials, and hands-on workshops. The instructor also provided some supplemental guides and tip sheets. (p. 30)

As described, Kalman & Ellis (2007) combine Web, print, and multimedia resources in an attempt to address students' differing learning styles and to utilize the Web as more than an 'electronic textbook'. Providing instructional materials in multiple media formats (e.g., Web-based multimedia with voice and animation or motion video, in addition to text) has been recognized as an effective method for addressing differing student learning preferences (Bransford et al., 2000; Lever-Duffy et al., 2005; Newby et al., 2006): "By using a variety of resource materials…and addressing various learning modalities (by presenting information orally, visually, and in writing), teachers can address differences in preferred learning styles and achievement levels" (Wiggins & McTighe, 2005, p. 219). This is the argument often used to promote the transition from textbooks to Web-based instructional materials: Web-based multimedia in particular can be visual, textual or auditory (or simultaneously all three), meeting the needs of multiple learning preferences in one locale (Lever-Duffy et al., 2005; Newby et al., 2006). Multimedia has also been recommended since "the ability of the human mind to quickly process and remember visual information suggests that concrete graphics and other visual representations of information can help people learn (Gordin and Pea, 1995)" (Bransford et al., 2000, p. 215).

Students can also ‘control’ Web-based multimedia in a way that encourages engagement (Zhang & von Dran, 2000). They may self-pace or even self-select sequencing for example, when studying Web-based videos (Lever-Duffy et al., 2005; Kahn, 2011). Also, the hypertext aspects of the Web mean that it is “a medium that enables learners to return to information resources they have examined before. In this way, they may keep examining the topic they are working on, and may expand that examination whenever they come across an associated topic” (Hazzan, 2004, p. 332). Multimedia has also been shown to “help learners more readily recognize meaningful prior learning and how the new information relates” (Newby et al., 2006, p. 33). This aligns with constructivist learning theory recommendations which promote “a developmental approach to learning where each child is seen as an active agent in his/her learning. The approach accords equal importance to what the child learns and to the process by which s/he learns” (Leahy & Twomey, 2005, p. 144). The potential for students to “individualize” their own learning and to identify their own studying needs is especially important for diverse groups of learners or groups with diverging entry competencies (Smaldino et al., 2005). It is still the instructor’s role to choose appropriate Web-based materials and media based on analysis of their particular student audience (Smaldino et al., 2005), but “in a well-designed [Web design class] the individual learner may be able to re-arrange the material that best suits his or her Cognitive Style, thus maximizing the potential of the medium” (Parkinson & Redmond, 2002, p. 42). Kahn (2011) also notes that students respond favorably to studying videos from Youtube.com, because it removes the pressure of another person asking “Do you understand?”—again, students were able to build their own understanding at their own pace, not the instructor's.

These arguments for the strengths of Web-based instructional videos are again tempered by non-technical students' lack of experience with self-instruction. But, as Mackey (2005) reports, Web-based materials can be implemented effectively in lecture-based courses where students' lack of skill can be otherwise addressed. Use of Web-based multimedia tutorials for example gains practice in Web design instruction, because it provides an "active learning opportunity for students to learn web design concepts on their own, in association with lecture materials, computer lab, and course readings" (Mackey, 2005, p. 3245). Mackey (2005) found that the "development of easy-to-understand web design tutorials became essential to [the] course design. Initially, the web based tutorials included text and image-only in static web pages, but these materials were replaced by multimedia tutorials that included screen-captures, video, and audio" (p. 3243). (This progression from text, to text and image, to video and audio is representative of how Web-based instructional materials evolve in general (Kahn, 2011).) Replacing text/image tutorials with video tutorials is not essential for Web design instruction—the choice should always be based on the learning, technical and content needs of the student/audience (Lever-Duffy et al., 2005; Mackey, 2005; Smaldino et al., 2005) and to foster meaningful learning—but expectations for using Web-based video as an instructional material increase as video becomes ever more prevalent on the Internet (Kahn, 2011). (A July 2012 keyword search for "learn HTML" returned over 60,000 video tutorials on YouTube.com alone. The result numbers are exponentially greater when using a general search engine like Google.com.) It should be noted that multimedia instructional materials do not meet the needs of all learners. For example,

> Smith and Woody…found that "the effects of multimedia teaching strategies do have some benefits but that these benefits do not fall evenly on all students" (2000). The impact of multimedia on student learning depends on the orientation of the student to be responsive to this approach (Smith and Woody, 2000). Specifically, the authors found that "those students who prefer visual input to verbal input will benefit more than those who are less visually oriented" (Smith and Woody, 2000). In addition, "verbally oriented students' performance does not improve, and may in fact suffer, as a consequence of the visual orientation of class presentations" (Smith and Woody, 2000). This suggests that the development of multimedia materials must consider design and functionality, as well as the specific profile and needs of the audience to engage with these materials. (Mackey, 2005, p. 3242)

The 'noise' of Web-based materials. Web-based multimedia tutorials should also be selected (or developed) to meet practical and educational standards such as promoting active learning, building on prior knowledge, and not overwhelming learners' memory with excessive visual or audio stimuli (Mackey, 2005). Multimedia's potential to 'overload' students' limited cognitive capacity has long been an argument for its restricted or selective use as an instructional material (Mayer & Moreno, 2003). Adding the visual and technological complexity of a Web page interface to instructional materials can compound the issue: "As a communications medium, the Web presents many challenges to effective message delivery not encountered in traditional instructional environments or in print publications" (Burch, 2001, p. 358). Not only is "reading text on a computer screen…more difficult and time consuming than reading printed text (Murphy, 1999)" (Burch, 2001, p. 362), but students are usually faced with studying Web pages that contain text, graphics, and multimedia, *plus* extensive amounts of website functionality like menus or (unfortunately) advertising—this "high level of noise" (Burch, 2001, p. 360) in a Web page interface can quickly lead to information overload (Hazzan, 2004). Cognitive load theory concludes that "unnecessarily forcing learners to work with disparate sources of mutually referring information leads to ineffective instruction and to increasing their cognitive load before the intended learning actually begins" (Doering & Veletsianos, 2007, p. 112)—in other words, the "noise in the user interface can be the most serious problem" (Burch, 2001, p. 365) related to using Web-based instructional materials, since it can lead to cognitive/information overload before a student even attempts the instructional content. Students can "only process a very small amount of new information at one time in working memory" and as such "information presented to learners should be designed in such a way as to reduce any avoidable load upon working memory" (Leahy, Chandler, & Sweller, 2003, p. 413). Students lacking experience with the Web may also 'get lost' navigationally or struggle with the "non-sequential and random gathering of information" (Hazzan, 2004, p. 325) that is common to the Web medium. For example, "as soon as a student clicks on a link that takes them offsite, the instructor loses control of the presentation of information. The instructor…risks losing the student completely" (Burch, 2001, p. 361). How experienced or even anxious students are with computer technology in general can then significantly influence how well they learn when using Web-based instructional materials (Chou & Wang, 1999).

Web design educators should not shrink from employing Web-based instructional materials though: in order to "meet the literacy needs of students [instruction] must effectively incorporate IT (web) concepts. As educators in a digital age we have a responsibility to provide students with the opportunity to engage with the web in all of its complexity" (Mackey & Ho, 2005, p. 554). Engaging with Web-based resources aids students in developing their information retrieval and processing skills (Lim et al., 2003), as well as offering them an "increased sense of user control, more task variety, less task routine, and provision for capabilities to move task performance to higher levels" (Zhang & von Dran, 2000, p. 1254). It is, after all, a goal of computer literacy education (and, as part of it, Web design education) that higher education students be "empowered users" who "although continuously challenged…integrate computers more productively and cope reasonably well in dynamic environments…confront skill demands, collaborate online, and explore instructional opportunities. In other words, they employ computers in order to further their educational goals" (Selber, 2004, p. 476). Web design students need to gain the conceptual experience/skills to "cope with the information surrounding them" (Hazzan, 2004, p. 339) and integrating Web-based resources and the tasks of constructing Web pages can address this (Lever-Duffy et al., 2005; Reber, 2005). Leahy and Twomey (2005) also argue that Web design instructors should consider employing Web-based materials, because "teachers in the schools of the future need to be open to the use of new technologies and have the capacity to adapt to the constantly changing learning environments in which they find themselves (OECD, 1998)" (p. 144). Instructors must also take into account that students likely have different technology expectations than they do:

> Recent surveys by the Pew Internet and American Life Project (Levin and.Arafeh, 2002) on the role of the Internet in schools suggest that there is a significant disconnect between teachers and their students about how the Internet fits with academics. Students report being hampered in their Internet use at school by teachers who tend to view the Internet as a resource to be consulted rather than as a tool for connecting with the world. (Tilley & Collison, 2007, p. 27)

Summary

What then does Web design instruction that incorporates the myriad of recommendations and instructional strategies look like? Teaching Web design's combination of computer, visual and information literacies is a complex endeavor, even more so when trying to instruct non-technical learners or novices with no prior knowledge on which to build. As described, there is little consensus on how to effectively teach Web design—the software, coding, programming or communication approach taken can vary depending simply on the academic field teaching the topic. But the common emphasis on procedural instruction, focusing on the finite processes needed to build Web pages rather than the conceptual understanding of Web design's abstractions, is proving to be insufficient: Web technology evolves quickly and frequently, and more than anything Web design students need to learn the concepts that allow them to adapt to changes (Kalman & Ellis, 2007).

It then falls to Web design instructors to reshape their pedagogy, to strive for a balance of conceptual and procedural learning, to impart an integrated understanding of

Web design theory and practice, as well as addressing the needs of novices. To follow the strategies, the instruction must teach not just content, but must also shift students towards becoming conceptual learners capable of self-instruction. The 'new theory of learning' (Bransford et al., 2000), with its strategies for learner-centered, knowledge-centered, assessment-centered, and community-centered instruction can be employed as a framework, as can strategies determined effective by modern learning theories like behaviorism, cognitivism and constructivism. In Web design instruction's scenario of not just teaching with the Web, but teaching the Web itself, numerous challenges not faced in traditional educational topics emerge—regardless, expectations that higher education students can and should create Web pages necessitate that instructors fully broach and understand how to effectively teach Web design (Karper, 2004).

And of equal importance: after the Web design instruction is developed, it must be rigorously analyzed and evaluated with "use-driven strategic research and development focused on issues of improving classroom learning and teaching" (Bransford et al., 2000, p. 250)—it is this purpose that this study uses as its rationale. In addition to evaluating the viability of the curriculum unit, the study seeks to better document which student conceptual misunderstandings persist and why, particularly for the understudied topic of CSS. The following questions are addressed by this study:

1. What deviations by the instructor occur during implementation of the curriculum unit and why?
2. What student 'misunderstandings' about XHTML and CSS persist throughout curriculum unit implementation and why?

The study uses as a framework Design-based Research (DBR)—DBR is grounded in both theory and real-world context, with an emphasis on innovation via instructional design, interactivity and flexibility. The goal of a design study is to "not only learn about learning…but also to support the development of particular forms of learning (thus contributing to students' knowledge)" (Reimann, 2011, p. 40)—it ultimately strives to "make learning research more relevant for classroom practices" (Reimann, 2011, p. 37). In keeping with this, when using DBR, "researchers assume the functions of both designers and researchers, drawing on procedures and methods from both fields" (Wang & Hannafin, 2005, p. 6).

Chapter Three

Methods

"The goal of DBR is to produce knowledge that will be useful in providing guidance to others in their attempts to support learning processes." (Reimann, 2010, p. 16)

This study investigated how to effectively teach Web design to non-technical learners, by evaluating a curriculum unit that was developed for teaching XHTML, CSS and Web design concepts to adult learners at the higher-education level. The study also strove to better document which student conceptual misunderstandings about XHTML and CSS persist and why, particularly for the understudied topic of Cascading Style Sheets—it was a continuation of and expansion on the investigation conducted during curriculum unit pilotings.

Curriculum Unit Development

The XHTML/CSS curriculum unit underwent extensive instructional design, development and piloting. It was originally developed using the ASSURE model of lesson planning found in *Instructional Technology and Media for Learning* (Smaldino et al., 2005). It was also evaluated and revised based on the "backward design" recommendations of Wiggins and McTighe in *Understanding by Design* (2005): The unit strives to craft understanding and transfer, not just information coverage. In particular, it emphasizes authentic performance stemming from authentic problems and constant formative assessment/tracking of students' 'misunderstandings.' Finally, the unit strives to incorporate the design tenets of the 'new theory of learning' put forth in *How People*

Learn: Brain, Mind, Experience, and School (Bransford et al., 2000). Learner-centered, knowledge-centered, assessment-centered and community-centered strategies were integrated throughout the lesson plans. The unit was designed to meet the needs of adult learners (college students) with limited or no prior experience creating Web pages. It was also developed to build students' experience with Web-based instructional materials (e.g, Web-based video tutorials, Web sites or online tools).

The curriculum unit consists of six lessons introducing the foundational concepts and coding practices needed to create Web pages (i.e, Web design). Each lesson follows a pattern of review/discussion, direct instruction/lecture, demonstration, and then practice. Assignments consist of building and composing Web pages in the computer languages XHTML and CSS. Students also read or watch Web-based tutorials or articles on XHTML, CSS and Web design concepts. The assignments continue the authentic 'practice' portion of each lesson; the Web-based instructional materials (e.g., videos and tutorials) aid students' self-directed learning outside the classroom. Some assignments also ask students to answer discussion questions in preparation for the in-class review sessions. Assessment consists of line-item feedback on the students' Web pages. The feedback presents students with recommendations for revision based on completeness, accuracy, and decision-making—students must then repair their Web page assignments based on the feedback.

External review panel. As precursor to the study, an external review panel was conducted on the curriculum. To establish content validity, "researchers go to a panel of judges or experts and have them identify whether the questions are valid" (Creswell, 2005, p. 165). (In this case, the lesson plans and assignments were reviewed for appropriateness for the audience and feasibility for implementation, not assessment questions. See Appendix A: Sample External Review Panel Questionnaire for an example of the review panel instrument.) Experts were chosen to serve on the panel based on their experience with the population of interest (adult learners lacking technical or Web experience). Reviewers were also chosen for their expertise with instructional design, teacher education or the content area. They were asked to gauge the curriculum unit's appropriateness for

a) implementation,
b) the target student audience (adult learners), and
c) the target instructor audience (post-secondary level instructors).

Reviewers were also asked to offer recommendations for improving the lesson plans. Revisions to the unit were then made based on the external review panel results and recommendations, e.g., examples and instructions in the lesson plans were expanded where requested and recommended strategies for improving student engagement were integrated.

Design & Rationale

To further the study's goal of evaluating the curriculum unit, the framing questions for the study were

1. What deviations by the instructor occur during implementation of the curriculum unit and why?
2. What student 'misunderstandings' about XHTML and CSS persist throughout curriculum unit implementation and why?

The curriculum unit was implemented for six weeks at the University of Virginia, and a mixed-method approach was used to investigate the study's research questions. The rationale for a mixed-method research design stemmed from the study's attempt to adequately evaluate and analyze the curriculum unit in an authentic or real-world context. During the implementation, the investigator served as instructor and participant-observer. (Independent/non-participant observers also collected data on the implementation using the observation protocol.) Document analysis of student data was completed to strengthen the conclusions drawn from the observation data (Creswell, 2005). The participant-observation, independent classroom observation and document analysis research methods used in the study were qualitative (Brogdan & Biklen, 2003); quantitative descriptive statistics and deductive qualitative analysis were used in reporting.

Quantitative statistical results collected from the observation protocol would not be sufficient to address the questions framing the study; qualitative methods and analysis were needed for in-depth exploration (Creswell, 2005). A mixed-method approach is also preferred, because "the combination of both [qualitative and quantitative] forms of data provides a better understanding of a research problem than one type of data alone" (Creswell, 2005, p. 53).

The study also followed a deductive approach, which

> starts deductively from pre-set aims and objectives. The data collection tends to be more structured than would be the norm for much other qualitative research and the analytical process tends to be more explicit and more strongly informed by a priori reasoning. (Pope, Ziebland, & Mays, 2000, p. 116)

The particular objective in this study was to evaluate the curriculum unit, i.e., the "model", referencing the empirical data on persistent student errors collected during pilotings—see Table 1: Student Misunderstanding Categories Compiled during Pilotings in chapter 1.

Design-based Research. A Design-based Research (DBR) framework was also chosen for the study: DBR is grounded in both theory and real-world context and in it, "researchers assume the functions of both designers and researchers, drawing on procedures and methods from both fields" (Wang & Hannafin, 2005, p. 6). In a design study, "an extended investigation of educational interactions [is] provoked by a set of designed, usually innovative, curricular tasks and/or educational technologies (Confrey, 2006)" (Reimann, 2011, p. 38). DBR is typically characterized as qualitative research that is "(a) pragmatic (i.e., design-oriented and intervention-oriented); (b) grounded in theory and research; (c) interactive, iterative and flexible; (d) integrative; and (e) contextual" (Reimann, 2011, p. 37). One of DBR's emphases is to "make learning research more relevant for classroom practices" (Reimann, 2011, p. 37), so that teaching and educational policies are directly impacted by evidence-based conclusions; it seeks to both "increase the relevancy of theory" in classrooms and to "increase the impact of best teaching practices on theories of learning" (Reimann, 2011, p. 39). DBR further aligned with this study's needs, because it "aspires to produce explanatory accounts that are not solely descriptive" (Reimann, 2011, p. 39), and because it supports deriving research findings from formative evaluations (Collins, 1992)—in this study's case, the observation protocol and the students' formative assessments. Also, a common outcome from DBR is a "design solution, such as a set of materials, tasks, and activities to teach a specific competence" (Reimann, 2011, p. 45), which further speaks to its frequent association with instructional design and educational technology research.

Site Description

The curriculum unit was implemented during a semester at the University of Virginia. The unit formed part of the syllabus of EDIS 7020: Courseware Design, a graduate-level instructional technology (IT) course offered by the Curry School of Education. EDIS 7020 explores the computer or Internet tools and technologies employed for instructional development—topics range from instructional design and interactive development, to educational multimedia, technology leadership and technology and teaching. (Topics vary by year.) EDIS 7020 met weekly for 2.75 hours in a computer lab on the University of Virginia campus. The Curry School of Education has approximately 1,800 students and 21 areas of study. (These Fall 2012 figures were reported on Curry.virginia.edu, "At a glance".)

The computer lab (classroom) where the course was taught was equipped with 9 PC computers, though all participants brought/used their personal laptops. The computer lab provided wireless Internet access. There was also an instructor's computer configured for use with a wall-mounted projector. Enrollment in the course was open to all Education majors and no prerequisite course was required.

EDIS 7020 was chosen as the implementation site because its students were adult learners from non-technical backgrounds (primarily K-12 education); they were the population of interest in this study and are the target learner-audience for the curriculum unit. A previous incarnation of the course (EDLF 702) was used as one of the piloting sites, and its viability as a study site that offered authenticity was determined then. The site was also chosen as a matter of convenience because of the researcher's existing relationship with its faculty instructor. The students' participation and assignment submission rates were reported to the faculty instructor, but did not impact their final grade.

Participants

A total of five graduate-level Education majors participated in the study. All participants were enrolled at the Curry School of Education at the University of Virginia and were adult learners, with prior degrees in Education. Four of the students were part of a first-year instructional technology cohort. The remaining student was an Educational Leadership, Foundations, and Policy major. Also, three of the five students held professional experience as K-12 instructors; a fourth was an online instructor and instructional materials developer for a test-preparation company. The fifth participant held professional experience as an educational consultant and instructional materials developer for a K-12 publishing company. None of the participants had taken a prior course in XHTML/CSS, Web design or Web design software, though three had attempted minimal self-instruction—they had looked up how to add a Web video to course materials, for example. Four of the five participants were currently posting content online using Web-based tools like learning management systems (LMS) or push-button publishing (e.g., social media interfaces). The students' prior knowledge of and experience with Web design and Internet technology was documented as part of the Lesson 1 assignment—it was entered on the students' assignment Web pages (see Table 2). Though two of the students did list prior experience with HTML and CSS, none of the

Table 2

Participants' Prior Experiences with Web Design, Web Publishing and Web-Based Instruction

Area	# Participants	Experiences
Web Design		
HTML	2	Push-button publishing; minimal self-instruction; created simple Web site using AngelFire interface in 1990's
CSS	2	Minimal self-instruction
Scripting	1	BASIC, MS-DOS, Java computer languages
WYSIWIG Software	2	Used Dreamweaver to revise existing work Web pages (self-instruction); Angelfire account
Posting Web Content	4	Push-button publishing: edited existing course Web site with Google Sites; LMS; blog posts; wiki updates; Twitter posts; social networking posts
Instructional Materials	3	Google Sites, LMS, CMS
Web-based Instruction		
Teaching	1	Instructor; materials developed for online test preparation courses
Learning Materials	5	Studied instructional Web pages: digital fabrication tutorials, video tutorials on HTML; Dreamweaver tutorials (Lynda.com); materials for HS classes
Enrolled in Online Course	1	
Web Account Admin	3	Google Sites, Wordpress blog, Twitter
Other Web-based Tools	2	Online photo editors; presentation tools like Prezi, Voice Thread; Popplet
Multimedia Development		
Video	4	iMovie, Camtasia, Windows Movie Maker, Hyperstudio software
Audio	2	Audacity software
Animation	3	Flash software
Graphic Design		
Photo/graphics	2	Photoshop, Illustrator, Superpaint software
Layout/print	2	Quark software; print design courses

participants knew how to build Web pages before the implementation, nor did they possess a conceptual understanding of Web pages and the Web design process.

Data Collection Methods

The XHTML/CSS curriculum unit was implemented in full in a graduate-level instructional technology course, with the researcher serving as instructor. Three data collection methods were used during the curriculum unit's implementation: classroom observation, video observation and document analysis of participants' assignments. For improved reliability and validity, independent, non-participant classroom observers were employed in addition to participant-observation, and all implementations were video-recorded for accuracy in reporting and evidence (Creswell, 2005). This "triangulation" or comparison of multiple data sources served to improve study validity (Creswell, 2005, p. 252).

Classroom observations. As participant-observer, the researcher self-assessed during each lesson plan implementation, as needed to function as instructor. An independent observer completed an observation protocol/checklist during each lesson, reporting the deviations from the lesson plans and student misunderstandings evident during the class—they denoted the instructor's behavior as well as students':

> An observational protocol is a form designed by the researcher before data collection that is used for taking fieldnotes during an observation. (Creswell, 2005, p. 223)

The observation instruments for this study were modified copies of each lesson plan that allowed deviations and student misunderstandings to be tracked and coded—see Appendix B: Sample Observation Instrument for an example. After each class, the researcher and independent observer debriefed, comparing impressions of the implementation and the effectiveness of the observation instruments (i.e., peer debriefing). All observers used the same observation instruments. Also, observers were trained in the instruments' use before implementation began. Both independent observers (Rater B, Rater C) were doctoral students at the Curry School of Education at the University of Virginia who had completed research methodology coursework. Rater B held degrees in computer science and instructional technology and had prior experience with performance analysis. Rater C held degrees in English education and had prior classroom observation and qualitative field-noting experience. Both were former curriculum unit pilot participants, so they were familiar with the project. The independent observers had no vested interest in the study results though (i.e., they had impartiality), for improved data collection credibility.

After the implementation of a lesson plan, the researcher reviewed the video recording of each lesson, field-noting and formally reporting on an observation instrument when areas in the lesson plans were covered, omitted or modified, and the reason why the deviation occurred. (The independent observers' instrument was also compared to the video recording, and any discrepancies in reporting were noted and investigated in order to confirm final codes.) The video recordings helped ensure accuracy in reporting and evidence (Creswell, 2005). Before each lesson implementation, the researcher set up the digital video camera in the back of the computer lab (classroom), so that all students, the instructor and the projector screen were visible.

Documents. Document analysis of students' assignments was also completed to strengthen and expand on the conclusions drawn from the observation data. The 49 documents reviewed were the participants' Web pages, their homework assignments, composed in the computer languages XHTML and CSS. (The students composed the documents in text-only or Web editing software.) While serving as instructor, the researcher completed line-item feedback on the students' assignments—this is the formative assessment that occurs during the curriculum unit. The feedback collected both conceptual misunderstandings and errors made in coding the computer languages and required close reading and analysis of each student's assignments. The document analysis also directly impacted lesson plan implementation (and thus, deviation), since the review sessions were reactive to students' difficulties with assignments. The students' Web pages afforded evidence of conceptual misunderstandings, and the reasons behind them, that could not be observed during the classroom implementation. The lesson plans themselves were also reviewed multiple times during analysis.

Data Management

The data corpus for this study includes video recordings of the curriculum unit implementation, observation protocols, students' assignments (HTML and CSS files), and the curriculum unit itself (the lesson plans, the website with assignment instructions, the demonstration Web pages, etc.). The study's research protocol was reviewed by the University of Virginia's Institutional Review Board (IRB) and approval granted. Participation in the study was voluntary and confidential—each participant was assigned a pseudonym to offer confidentiality, plus contact/identifying information was removed from any participants' Web pages used as examples in the study's results or discussion sections. Steps were taken at all times to honor participants' privacy, but true anonymity cannot be guaranteed, due to the nature of the curriculum unit and the assignments, i.e., students posted Web pages to the Internet, though they were in a secure location. Plus, the curriculum unit (and study) did not ask participants to post sensitive information, nor did it encourage student vulnerability via competition or coercion (Bogdan & Biklen, 2003). The students' class participation and assignment submission rates were reported to the study site's faculty instructor, but again, the unit was only six weeks of a 16-week course, and the unit did not impact student's final course grades. Students enrolled in the course who declined to participate in the study were led in alternative activities by the faculty instructor.

The video recordings and coding keys for pseudonyms or analysis were kept on a password-protected laptop at all times.

Data Analysis

This study practiced deductive analysis of data, where "the objectives of the investigation [are] typically set in advance" (Pope et al., 2000, p. 116), rather than emerging from the data or literature (i.e., induction).

> In deductive qualitative analysis, researchers begin their research with a conceptual model, study cases in depth, and then reformulate the model to fit the cases. A key procedure in deductive qualitative analysis is the active search for evidence that undermines the current conceptual model. (Gilgun, 2005, p. 83)

In this study's scenario, the curriculum unit and its instructional methods/strategies were the 'model' being tested or reformulated; the deviations and student misconceptions that occurred during implementation were the undermining evidence. The study also referenced the empirical data on persistent student errors documented during pilotings—see Table 1 in chapter 1. In other words, the structure of analysis was operationalized on the basis of previous knowledge as is common with deductive analysis (Kyngas & Vanhanen, 1999). Also, in the DBR framework, "making sense of the data is typically a highly inferential, interpretative, and cyclical process" (Reimann, 2011, p. 42).

Data analysis began with coding of the deviations reported on the observation protocols (instruments). The observation instruments in this study were adapted copies of each lesson plan that allow for detailed coding and field noting to be reported during lesson plan implementation (or video review of). Each lesson plan's content was converted into a numbered checklist and divided into subsections to better aid coding and analysis—see Appendix B: Sample Observation Instrument as example. A simple coding scheme was used to report the observation protocol results—Covered, Modified, Returned to, Omitted, Not Discernible—and to generate descriptive statistics that "summarize the overall trends and tendencies in [the] data" (Creswell, 2005, 182). Student questions asked during implementation that impacted deviations were also itemized and coded during the video review, in order to track their impact on deviations. (Again, the recurring student misconceptions collected during piloting served as the coding categories.) The participant-observer's and independent observers' instruments were also compared for agreement/conflict in reporting; the video was reviewed multiple times for final determination of codes, as well as data trends and frequency. An interrater reliability analysis using Cohen's Kappa was completed to determine consistency among raters. The observation data was then qualitatively analyzed as the results were interpreted:

> Deductive qualitative analysis is different from grounded theory in its use of preliminary theory and coding, but eventually the procedures of both approaches converge in data analysis and in the writing up of results. (Gilgun, 2011, p. 2)

Data analysis proceeded with content analysis of student documents (assignments)—they were read multiple times and categorized/coded for comparison to the observation data. Coding in document analysis "is used to break up and segment the data into simpler, general categories *and* is used to expand and tease out the data, in order to formulate new questions and levels of interpretation" (Coffey & Atkinson, 1996, p. 30). On the first pass of analysis, the student misunderstandings and errors reported in the feedback comments were itemized and coded. On additional readings, the document analysis was used to gather insight for interpreting the patterns and themes evident in the observation data (Marshall & Rossman, 1999).

Study Validity

To address the weaknesses or biases associated with the study's research methods—particularly participant-observation—multiple steps were taken to improve the study's validity and objectivity.

> A common criticism directed at qualitative research is that it fails to adhere to canons of reliability and validity (LeCompte & Goetz, 1982). Internal validity is concerned with how trustworthy the conclusions are that are drawn from the data and the match of these conclusions with reality, while external validity refers to how well conclusions can be generalized to a larger population. (Anfara, Brown & Mangione, 2002, p. 33)

Put another way:

> Validity deals with the notion that what you say you have observed is, in fact, what really happened. In the final analysis, validity is always about truth. (Shank, 2002, p. 92)

The study's mixed-method design—e.g., using both quantitative and qualitative reporting and multiple data collection methods—was its first step towards improving validity; "a basic rationale for [mixed method] design is that one data collection form supplies strengths to offset the weaknesses of the other form" (Creswell, 2005, 514). In addition, qualitative triangulation was used throughout the study:

> Triangulation is the process of corroborating evidence from different individuals (e.g., a principal and a student), types of data (e.g., observational fieldnotes and interviews), or methods of data collection (e.g., documents and interviews) in descriptions and themes in qualitative research. (Creswell, 2005, p. 252)

For example, employing independent observers and video recordings of the curriculum unit implementation offered improved objectivity during the data collection and management processes. Training the independent observers to use the observation protocol was enacted to further strengthened data credibility (Creswell, 2005).

The study's external review panel also served to improve the study's content validity (Creswell, 2005). Again, reviewers were chosen for their expertise with the population of interest, and they completed questionnaires gauging the appropriateness of the lesson plans for implementation with the student and instructor audiences for whom the curriculum unit was designed. (See Appendix A: Sample External Review Panel

Questionnaire for an example.) Revisions to the curriculum unit, based on the review panel's feedback, were made prior to implementation at the study site.

The subjectivity inherent to participant-observation and classroom observation methods, i.e., data collector bias, remained a concrete threat to validity in this study: "the qualitative researcher's challenge is to demonstrate that [their] personal interest will not bias the study" (Marshall & Rossman, 1999, p. 28). The use of the video recordings plus non-participant, impartial observers for evidence gathering again attempted to address this threat. The use of both quantitative reporting and qualitative analysis techniques also attempted to alleviate opportunity for bias in analysis, as did use of "multiple levels of analysis" (Creswell, 2005, p. 274).

Additionally, the presence of the researcher and observers likely altered the behavior of the study participants; they served as reminders that participants were being observed. This is known as the Researcher Effect or Heisenberg Effect (Brogan & Biklen, 2003). The knowledge that they are being video-recorded may also have led to some discomfort. To address these threats to validity, participants' were clearly notified of all data collection methods, including video recording, prior to the study. (Only the researcher had access to the video for review.) Pseudonyms for participants were used in all reporting to honor their privacy, and identifying information was removed from Web pages used as examples during the study's results and discussion sections.

Researcher as Instrument

As the researcher, I functioned as participant-observer, conducting and present for all study activities. As instructor and curriculum developer, I taught the curriculum unit—this meant serving as teaching assistant for the authentic study site (EDIS 7020:

Courseware Design). Each lesson followed a pattern of review/discussion, direct instruction/lecture, demonstration, and then practice—I oversaw each portion of this process, concurrent with coordinating data collection. As the researcher, I also conducted all analysis and reporting of findings. (I also conducted the external review panel and independent-observer training, where they learned to use the observation protocol/checklist.)

I must acknowledge my potential for bias and for generating "personal theory" unsupported by evidence (Marshall & Rossman, 1999, p. 29) during this study, stemming from my involvement in all portions of the study's design and implementation. I must also be aware of how my prior experience studying the curriculum unit, as its designer and developer, may have been detrimental to objectivity in data collection and analysis. The nature of the study's analysis—deductive qualitative analysis which strives to 'undermine' the model or curriculum unit (Gilgun, 2005)—strove to address this potential for bias. The multiple data collection methods were also chosen to alleviate opportunity for bias.

I must also acknowledge my potential bias in the role as instructor and subject matter expert. During curriculum unit development, conscious effort was devoted to avoiding an Expert Blind Spot. Meaning, the curriculum unit was designed to meet the needs of novices, with the understanding that experts and novices have different cognitive and instructional needs (Bransford et al., 2000; Wiggins & McTighe, 2005). (The external review panel evaluated the curriculum unit's appropriateness for novices and non-technical learners, determining that it was appropriate.) However, I still brought my prior experience as a professional Web designer and developer to the study's instruction and implementation and a re-imposition of my Expert Blind Spot during classroom interactions was possible. Subject matter expertise is crucial for the success of any curriculum unit (Bransford et al., 2000); it is especially so for this unit, where the instructor must be able to identify errors and perform problem-solving in multiple computer languages at once. Regardless, I must maintain awareness of my Expert Blind Spot to avoid potential bias.

Summary

This study was designed to evaluate and analyze the XHTML/CSS curriculum unit in an authentic setting featuring the population of interest: adult, non-technical learners. The data collection methods—participant-observation, classroom observations, document analysis—were chosen to offer extensive data that affords multiple levels of analysis and improved study reliability and validity via triangulation (Creswell, 2005). The mixed-method study design also employed both quantitative reporting of the information gathered on the observation protocols, coupled with deductive qualitative analysis procedures. There were multiple threats to study validity and potentials for bias in this study's design—due primarily to the researcher serving in multiple roles including instructor and participant-observer—but the video recording of observations, use of impartial, independent observers, and an external review panel attempted to alleviate these threats by providing objectivity and credibility in reporting and evidence (Creswell, 2005).

Chapter Four

Results

"A key procedure in deductive qualitative analysis is the active search for evidence that undermines the current conceptual model." (Gilgun, 2005, p. 83)

Introduction

This study examined how best to teach Web design to non-technical learners, by evaluating a curriculum unit that was developed for teaching XHTML, CSS and Web design conceptually to adult learners. Of particular interest were novices' persistent misunderstandings and the impact of these on deviations during the curriculum unit's implementation. A mixed-method research design of classroom observations, video observations and document analysis was used to investigate the study's research questions:

1. What deviations by the instructor occur during implementation of the curriculum unit and why?
2. What student 'misunderstandings' about XHTML and CSS persist throughout curriculum unit implementation and why?

The study and its research questions were a continuation of an investigation begun during curriculum unit piloting. Therefore, a deductive approach and deductive qualitative analysis were used. In this methodology, the study

> starts deductively from pre-set aims and objectives. The data collection tends to be more structured than would be the norm for much other qualitative research

> and the analytical process tends to be more explicit and more strongly informed by a priori reasoning… (Pope, Ziebland, & Mays, 2000, p. 116)

For example, coding of student misunderstandings was based on data collected and categorized during lesson plan pilotings—see Table 7. A Design-based Research (DBR) framework also guided the study. DBR is characterized as qualitative research that is iterative, pragmatic, uses both theory and research and “aspires to produce explanatory accounts that are not solely descriptive” (Reimann, 2011, p. 39), as well as “make learning research more relevant for classroom practices” (Reimann, 2011, p. 37).

The results of the data collection and analysis as they relate to each aspect of the research questions are reported in this chapter, first via quantitative reporting (descriptive statistics which summarize the trends and frequencies in the data) and second, through key findings on the reasons for deviation and student misunderstandings during curriculum unit implementation.

Observation Data

Using an observation protocol—modified lessons plans converted into checklists for coding and fieldnoting; see Appendix B: Sample Observation Instrument—deviations from the lesson plans during implementation were coded into six categories: Covered, Modified, Returned to, Replaced, Omitted, and Not Discernible (see Table 3). Independent classroom observers coded deviations during each lesson, while the participant-observer watched video recordings of each class afterwards. In total, there were six lessons with 58 sections/topics coded (see Table 4). To make the final determination of deviation codes, observer disagreements were resolved by reviewing the video multiple times—see Interrater Reliability in this chapter.

Table 3

Deviations/Observations - Code Key

Code	Deviation	Description
COV	Covered	Section, topic or activities were addressed to the extent (and sequence) indicated in the lesson plan. Paraphrasing was likely and OK.
MOD	Modified	Section or topic was addressed but the explanation or activity was expanded or reduced, e.g., the examples were altered, dropped or more were used.
RET	Returned	Section or topic was addressed after being skipped, was reviewed twice or more, or was addressed out of sequence.
MREP	Replaced	Topic or activities were replaced with information or strategies not in the lesson plan. E.g., lecture wasn't working, so instructor replaced it with demonstration.
OMIT	Omitted	Section or topic was left out and not returned to.
NOTD	Not Discernable	Section or topic could not be observed. Class ended or observer's view or hearing was obstructed.

Table 4

Deviations (N=58) during Curriculum Unit Implementation

	Lesson													
	1		2		3		4		5		6		Unit Total	
Deviation	*n*	%	*n*	%	*n*	%	*n*	%	*n*	%	*n*	%	*n*	%
COV	2	16.7	3	23.1	3	42.9	4	36.4	2	28.6	5	62.5	19	32.8
MOD	6	50	7	53.8	2	28.6	4	36.4	3	42.9	2	25	24	41.4
RET	3	25	1	7.7	1	14.3	2	18.2	–	–	1	12.5	8	13.8
MREP	–	–	1	7.7	–	–	–	–	1	14.3	–	–	2	3.4
OMIT	1	8.3	–	–	–	–	–	–	1	14.3	–	–	2	3.4
NOTD	–	–	1	7.7	1	14.3	1	9.1	–	–	–	–	3	5.2
Lesson Total	12	100	13	100	7	100	11	100	7	100	8	100	58	100

For the unit *(N=58)*, the most frequent deviation observed was Modified (*n*=24), while Replaced (*n*=2) and Omitted (*n*=2) were least frequent. Each lesson had a separate number of sections, ranging from seven to 13 in number, and therefore a different number of codes observed. To determine which lessons underwent the most change overall, sections coded as Covered (*n*=19) were excluded (see Figure 4). Total non-COV deviation percents ranged from a high of 83.3% for Lesson 1 to a low of 37.5% for lesson 6, with a standard deviation of 16.4.

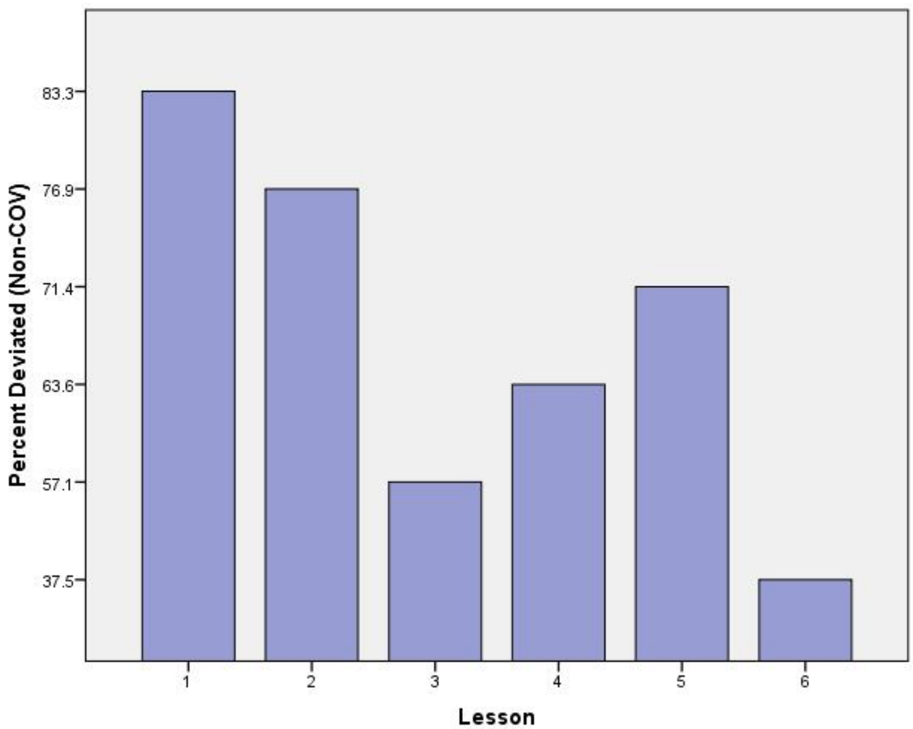

Figure 4. Percent of non-COV deviations by lesson, i.e., percent total that each lesson changed.

Reasons for deviations. Analysis of observation data revealed five reasons for the deviations during lesson plan implementation: Student Questions, Expansion/ Strategy, Class Ended, Instructor Error and Lesson Plan Defect (see Table 5). For the 39 sections with non-COV deviation codes, 44 reasons for deviation were coded during

Table 5

Reasons for Deviation - Code Key

Code	Reason	Description
STUQ	Student Questions	Students asked questions which altered the content or sequence of a section or topic, e.g., during demonstration, students' questions prompted the instructor to return to an earlier topic and cover it again.
EXPS	Expansion/ Strategy	Instructor expanded the explanation or used a topic as an example or demonstration in another section. The deviation was a strategic decision by the instructor.
ENDC	Class Ended	Section/topic was omitted or addressed without detail because class ended.
INST	Instructor Error	Instructor skipped a section or topic by accident. Instructor's incomplete explanation of a topic or poor paraphrasing led to student confusion, which prompted further deviation.
LESS	Lesson Plan Defect	Lesson plan lacked adequate written instructions. Instructions for implementation were unclear to observers. Artificial separation of sections led to deviation.

Table 6

Reasons for Deviation (N=44)

	Lesson												Unit Total	
	1[a]		2[b]		3[c]		4		5		6			
Reason	*n*	%	*n*	%	*n*	%	*n*	%	*n*	%	*n*	%	*n*	%
STUQ	4	30.8	5	45.5	2	40	4	57.1	–	–	–	–	15	34.1
EXPS	3	23.1	2	18.2	1	20	1	14.3	2	40	2	66.7	11	25
ENDC	–	–	1	9.1	1	20	1	14.3	3	60	1	33.3	7	15.9
INST	4	30.8	1	9.1	–	–	1	14.3	–	–	–	–	6	13.6
LESS	2	15.4	2	18.2	1	20	–	–	–	–	–	–	5	11.4
Lesson Total	13	100	11	100	5	100	7	100	5	100	3	100	44	100

[a] Lesson 1 contained three sections with dual codes (one with STUQ+EXPS, two with STUQ+INST). [b] Lesson 2 contained one section with dual codes (STUQ+LESS). [c] Lesson 3 contained one section with dual codes (STUQ+LESS).

observation (see Table 6). Five of the 39 sections contained two reason codes; STUQ plus one other reason code were observed (STUQ+INST, STUQ+EXPS, STUQ+ LESS). For the unit (*N*=44), the most frequent reason for deviation observed was Student Questions (*n*=15), while Lesson Plan Defect (*n*=5) was least frequent (see Figure 5).

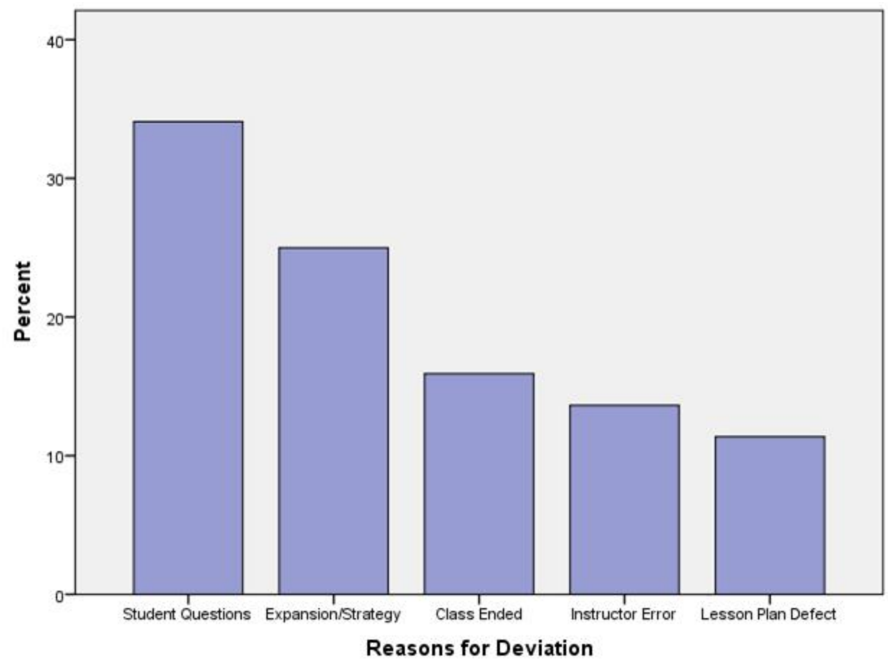

Figure 5. Reasons for deviation from the lesson plans (unit totals by percent).

Student questions in class. Because observation data revealed student questions and misunderstandings as the most prevalent reason for deviation during implementation (STUQ=34.1%; see Table 6), the questions students asked in class were also collected during observation and then coded by student misunderstanding (see Table 8). For the unit (*N*=248), the most frequent question observed concerned HTML Pathing (HPAT, *n*=36), while questions about coding accuracy (TYPO, *n*=5) were least frequent (see Figure 6). Multiple students asked questions from multiple categories during each lesson and misunderstandings were shared by students regardless of prior experience.

Table 7

Student Misunderstandings/Errors - Code Key

Code	Misunderstood Concept	Example
CCAS	CSS styles cascade from outer rules to inner/lower rules	*background-color* set for the body tag is duplicated in a lower rule needlessly; *text-align: center* isn't needed on links because the .navmenu class is already centered
CCON	CSS only styles content present in the HTML	*#leftcolumn* rule added in the CSS, but missing *id= "leftcolumn"* in HTML; CSS rules added for but tag not present in the HTML
CDIS	CSS translates to display	*text-align:left* added when it was already displayed by default; *list-style-type: circle* doesn't display b/c menu style is set in a different rule; *clear: both* applied to .intro class has no impact on display; LI *margin: 20px* is overridden when the A nested inside is set to *margin: 100px*
CPRP	CSS properties have specific functions and pair with specific values	nonexistent property/value was used (e.g., *float: bottom* doesn't exist; *float: center* doesn't exist and was used instead of *margin: auto)*; mismatched property/value pair was used (e.g., different properties use different keyword values like *medium*, *bold* or *none*; others use measurement values like *px*, *%* and *em*); duplicate *font-family* property/ value was used in the same rule
CSYN	CSS syntax has punctuation, grammar and shorthand	semicolons missing; *font-family* missing values in the chain; *border: red* used when *border-color: red* is correct; grouping syntax not used (e.g., *h2*, *#leftcolumn h2*); *margin: 30px* and *margin-bottom: 30px* shorthand conflict in the same rule
CVSH	Style and content are separate (CSS vs. HTML)	<strong> tag used instead of CSS *font-weight: bold*; <b> tag used instead of <span> and CSS *font-weight: bold*
CWID	CSS layout is created using float, width, clear, display, margin and padding	width, float and clear properties were not added together, so columns display incorrectly; *margin: auto* and *text-align: center* were used interchangeably, which they are not; *float:left* was used when *display: inline* was correct
HDIS	HTML tags translate to display	Tags open/close in the wrong place (e.g., is placed at the sentence end, instead of after the word); <strong> tag is nested inside <h1> tag, i.e., no understanding of default heading display; <h1> tag is before the <div> column for the content it labels, when it should display inside
HDIV	DIV/span are only used when existing tags are insufficient	<div class="bottom"> tag was added, but since only one <h3> tag inside, *div* can be deleted and the class reassigned to the *h3* (e.g., <h3 class="bottom">)
HIDC	ID and CLASS describe and differentiate tags	*id="container"* used twice, when ID names may only be used once; separate class names were added for *div*s with shared styles
HNES	Tags must nest properly	<title> not nested in <head>; <h1> nested inside an , which is not allowed, i.e., no understanding of how to nest list tags
HPAT	Pathing to URLs, files, and images must be exact	Broken links or display because image, CSS or HTML files uploaded to the wrong location on the Web server; full-path used instead of relative path to an image in the local folder; spaces and punctuation used in directory names and link paths (e.g., ../WHERE%20can%20tools?/)

HSEM	Tags are added semantically, based on meaning	<p> <big> used instead of <h1>, i.e., confusion between text title tag semantics; text isn't inside tags (e.g., missing or <dt><dd> tags); <link> to CSS is placed inside <body>, not <head>; <div> is nested inside <dl> when <dd> should be used
HSYN	HTML tags have syntax, punctuation, pairing and mirroring	Closing </p> tags not added; added </img> instead of closing slash in unpaired tag; Non-existant or deprecated tags used; HTML special characters not used
TYPO	Coding must be accurate and precise	HTML and CSS typos; *em2* used as value instead of *2em*; space between # and hex color prevents color from displaying; semicolon used instead of colon; extra space in *width: 20 px* breaks display; *ref* used instead of *href*; random </p> and </p></p> are likely copy/paste errors

Table 8

Student Questions Asked in Class (N=248) by Misunderstanding/Error

	Lesson													
	1		2		3		4		5		6		Unit Total	
Questions	*n*	%	*n*	%	*n*	%	*n*	%	*n*	%	*n*	%	*n*	%
CCAS	–	–	–	–	3	7.3	2	5.9	2	3.1	1	2.1	8	3.2
CCON	–	–	–	–	1	2.4	–	–	2	3.1	1	2.1	4	1.6
CDIS	–	–	1	3.6	3	7.3	6	17.6	8	12.5	9	19.1	27	10.9
CPRP	–	–	–	–	–	–	2	5.9	1	1.6	1	2.1	4	1.6
CSYN	–	–	–	–	4	9.8	7	20.6	8	12.5	6	12.8	25	10.1
CVSH	–	–	4	14.3	6	14.6	3	8.8	3	4.7	5	10.6	21	8.5
CWID	1	2.9	–	–	–	–	4	11.8	16	25	8	17	29	11.7
HDIS	5	14.7	4	14.3	3	7.3	3	8.8	6	9.4	4	8.5	25	10.1
HDIV	1	2.9	–	–	1	2.4	2	5.9	8	12.5	2	4.3	14	5.6
HIDC	–	–	–	–	1	2.4	–	–	5	7.8	1	2.1	7	2.8
HNES	2	5.9	–	–	–	–	2	5.9	2	3.1	1	2.1	7	2.8
HPAT	10	29.4	10	35.7	11	26.8	–	–	1	1.6	4	8.5	36	14.5
HSEM	4	11.8	3	10.7	4	9.8	–	–	1	1.6	3	6.4	15	6
HSYN	10	29.4	6	21.4	4	9.8	3	8.8	1	1.6	1	2.1	25	10.1
TYPO	1	2.9	–	–	–	–	–	–	–	–	–	–	1	0.4
Lesson Total	34	100	28	100	41	100	34	100	64	100	47	100	248	100

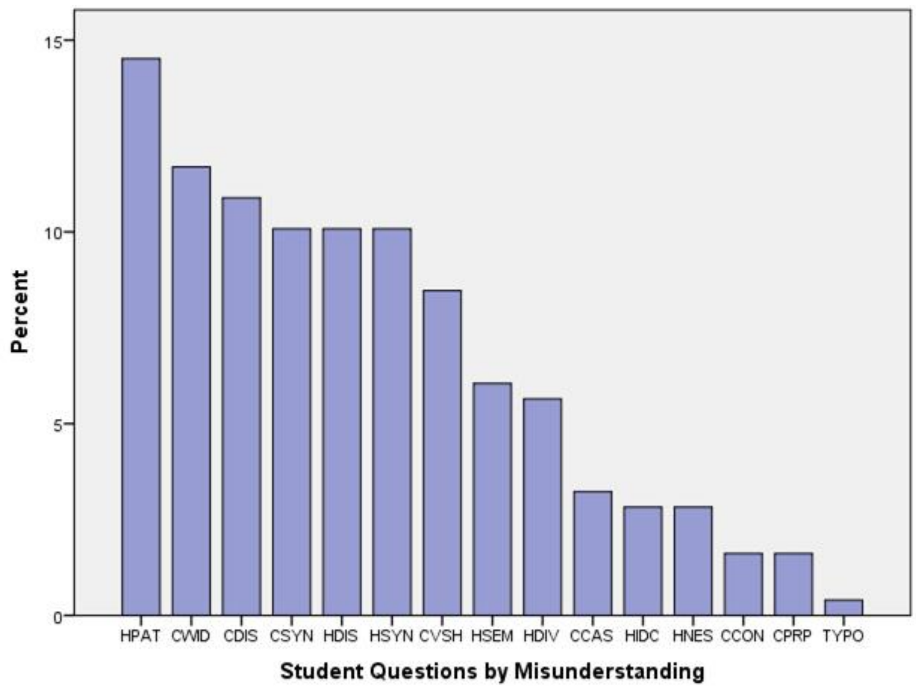

Figure 6. Student questions asked in class by misunderstanding (unit totals by percent).

Technical difficulties. Because the concept of HTML pathing generated the most student questions during class (HPAT=14.5%), the technical difficulties that occurred during class were also collected from the observation data (see Table 10). File pathing is not solely an HTML or Web design concept; it is actually a function of computer operating systems and a computer literacy concept. Therefore, technical difficulties encountered during lesson plan implementation with computer files, software and hardware were coded into six categories: File Management, Web Server, HTML Editor, Browser, Computer OS and Course Web Page (see Table 9).

For the unit *(N=39)*, the most frequent technical difficulty observed was File Management (*n*=18), while Computer OS (*n*=2) and Course Web Page (*n*=2) difficulty were least frequent (see Figure 7). Both the students and instructor experienced file management difficulty during implementation. Technical difficulties were not identified

Table 9

Technical Difficulties - Code Key

Code	Difficulty	Description
FILE	File Management	Difficulty locating HTML, CSS or image files; multiple copies of file open at once; unable to differentiate between file names; students saved over files by accident.
SERV	Web Server	Confusion between the local computer and the remote Web server, e.g., no understanding of the /public/ folder; difficulty using Dropbox file storage.
EDIT	HTML Editor	Difficulty operating the HTML or Web editing software (Kompozer, Sublime Text or Adobe Dreamweaver); Mac vs PC software differences caused confusion.
BRWS	Browser	Confusion on Internet browser functionality, e.g., no understanding of why feedback code does not display in browsers; no understanding that browsers display code differently.
OSPC	Computer OS	Laptop disconnects from Internet and student delayed while it rebooted; Mac vs PC operating system (OS) functionality caused confusion.
CORS	Course Web Page	Difficulty locating assignment URL; difficulty using online LMS, Coursekit.

Table 10

Technical Difficulties in Class (N=39)

	Lesson													
	1		2		3		4		5		6		Unit Total	
Difficulty	*n*	%	*n*	%	*n*	%	*n*	%	*n*	%	*n*	%	*n*	%
FILE	3	37.5	3	42.9	3	42.9	3	50	3	60	3	50	18	46.2
SERV	1	12.5	2	28.6	2	28.6	1	16.7	–	–	1	16.7	7	17.9
EDIT	2	25	1	14.3	2	28.6	–	–	1	20	–	–	6	15.4
BRWS	–	–	1	14.3	–	–	2	33.3	1	20	–	–	4	10.3
OSPC	1	12.5	–	–	–	–	–	–	–	–	1	16.7	2	5.1
CORS	1	12.5	–	–	–	–	–	–	–	–	1	16.7	2	5.1
Lesson Total	8	100	7	100	7	100	6	100	5	100	6	100	39	100

as an exclusive reason for deviation during implementation, though (see Table 5). They caused confusion (as evidenced by student questions) as well as delays while the instructor helped students locate files or problem-solve software issues, but observation data revealed that the instructor resumed the lesson plan content and sequence after each technical difficulty delay.

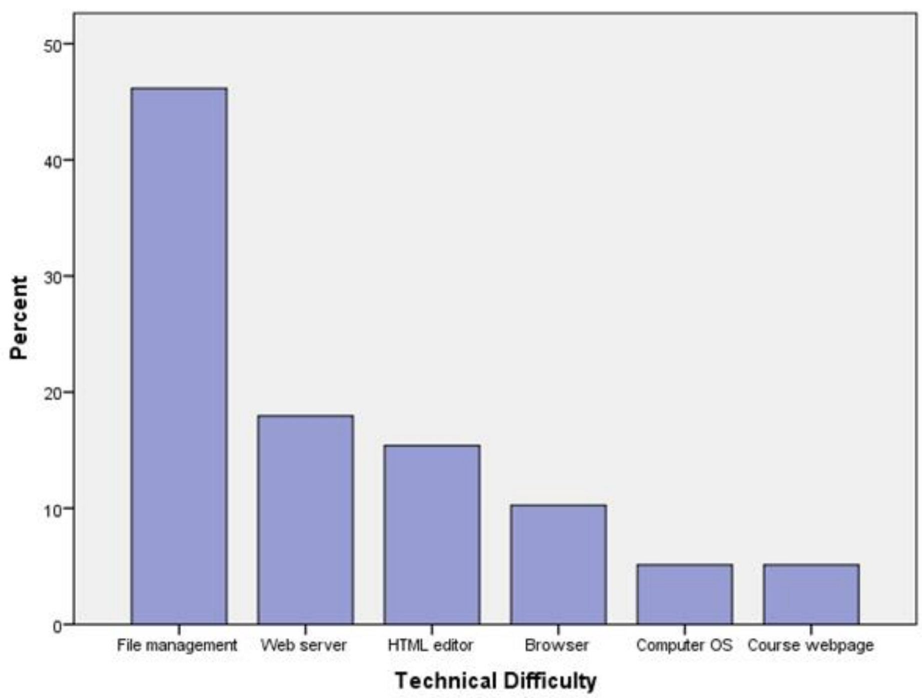

Figure 7. Technical difficulties encountered in class (unit totals by percent).

Interrater reliability (Cohen's Kappa). The participant-observer (Rater A) and two independent classroom observers (Rater B, Rater C) collected the observation data. Rater A and B observed lessons 1, 3, and 5 with a total of 26 deviations coded (see Table 11). Rater A and C observed lessons 2, 4, and 6, with 32 total deviations coded (see Table 12). An interrater reliability analysis using Cohen's Kappa was completed to determine consistency among raters. The interrater reliability for raters A and B was found to be Kappa = 0.67 ($p < .0.001$), 95% confidence interval (0.416, 0.876). The interrater reliability for raters A and C was found to be Kappa = 0.63 (p <.0.001), 95% confidence interval (0.404, 0.816). Both Kappa results may be interpreted as “substantial agreement” between raters based on Landis and Koch’s (1977) scale and as “fair to good” agreement based on the scale by Fleiss, Levin and Paik (2013).

Table 11

*Rater B * Rater A - Deviation Code Frequencies for Lessons 1, 3 and 5*

	Rater A						
Rater B	COV	MOD	RET	MREP	OMIT	NOTD	Total
COV	5	–	–	–	–	–	5
MOD	1	10	3	–	–	–	14
RET	–	–	1	–	–	–	1
MREP	–	–	–	1	–	–	1
OMIT	–	–	–	–	2	–	2
NOTD	1	1	–	–	–	1	3
Total	7	11	4	1	2	1	26

Table 12

*Rater C * Rater A – Deviation Code Frequencies for Lessons 2, 4 and 6*

	Rater A						
Rater C	COV	MOD	RET	MREP	OMIT	NOTD	Total
COV	9	1	–	1	–	–	11
MOD	3	11	–	–	–	–	14
RET	–	–	2	–	–	–	2
MREP	–	1	–	–	–	–	1
OMIT	–	–	1	–	–	–	1
NOTD	–	–	1	–	–	2	3
Total	12	13	4	1	0	2	32

Observer disagreement on codes was resolved by reviewing the video of the curriculum unit implementation multiple times. Cumulatively, rater A and rater B disagreed on six section codes, while rater A and rater C disagreed on eight. Analysis of the observation data revealed four trends behind the 14 total disagreements:

1. Raters disagreed on the extent to which the content was covered (COV) or modified (MOD) (n=5).
2. The classroom observer (rater B or C) did not revise earlier codes to RET when a section/topic was Returned to (n=5).
3. Rater B coded two sections as Not Discernable (NOTD), while video observation by Rater A indicated that they were Covered (COV) or Modified (MOD) (n=2).
4. Rater A and Rater C disagreed on two sections in Lesson 2 due to different interpretations of the lesson plan instructions (n=2).

Document Analysis Data

A total of 49 documents were reviewed and analyzed. These documents consisted of the students' XHTML and CSS files, i.e., their homework assignments, which they submitted during curriculum unit implementation. Because they worked on different website projects, each student submitted a separate number of files. Only submitted documents on which line-item feedback (formative assessment) was provided by the instructor are reported here. (In-progress drafts about which participants asked questions in class were excluded.) All errors and conceptual misunderstandings itemized during the feedback process where coded using the student misunderstanding/error categories identified during pilotings (see Table 1 in chapter 1). All but one of the misunderstanding categories appeared in the study's documents, and no new error categories emerged. No current participants made the error of using a MAC-only font within their assignments—that error code was dropped out. Also, in pilotings, subcategories within misunderstandings were assigned separate codes based on high frequency, but in this study 13 subcategories were re-incorporated into their umbrella categories based on low frequency, meaning that only 15 total categories are reported here. For example, out of the 275 total errors, student misunderstandings of the concept of HTML special characters were documented only six times. Because the concept of special characters is a subcategory of the larger concept of HTML syntax, these six errors were reintegrated and coded as part of HTML syntax (HSYN). Of the 275 total misunderstandings coded during document analysis (see Table 13), unique errors totaled 224, while recurring errors totaled 51. Recurring errors were either

a) student mistakes carried over from a prior assignment to a subsequent assignment, typically because the student failed to correctly complete/ understand the revisions based on feedback (a part of each homework assignment) or because the student copy/pasted erroneous code from one document to another, or
b) the same error made multiple times within the same document, e.g., the student typed *border: inset* in the CSS twice, when *border-style: inset* is correct, representing one misunderstanding recurring within the same document, not two separate misunderstandings.

For the unit *(N=275)*, the most frequent misunderstanding/error present in the homework assignments related to HTML syntax (HSYN, *n*=43), while CSS layout errors (CWID, *n*=5) were least frequent (see Figure 8).

Table 13

Misunderstandings/Errors (N=275) in Homework Assignments (Web Pages)

	Lesson												Unit Total	
	1		2		3		4		5[a]		6[b]			
Errors	*n*	%	*n*	%	*n*	%	*n*	%	*n*	%	*n*	%	*n*	%
CCAS	–	–	–	–	–	–	9	8.3	1	2.9	–	–	10	3.6
CCON	–	–	–	–	2	5.4	5	4.6	2	5.9	–	–	9	3.3
CDIS	–	–	–	–	1	2.7	12	11.1	1	2.9	–	–	14	5.1
CPRP	–	–	–	–	1	2.7	15	13.9	6	17.6	–	–	22	8
CSYN	–	–	–	–	10	27	15	13.9	4	11.8	–	–	29	10.5
CVSH	–	–	–	–	6	16.2	6	5.6	1	2.9	–	–	13	4.7
CWID	–	–	–	–	–	–	3	2.8	2	5.9	–	–	5	1.8
HDIS	3	7.5	4	7.5	1	2.7	–	–	–	–	–	–	8	2.9
HDIV	–	–	–	–	–	–	6	5.6	2	5.9	–	–	8	2.9
HIDC	–	–	–	–	–	–	5	4.6	2	5.9	–	–	7	2.5
HNES	5	12.5	11	20.8	1	2.7	4	3.7	1	2.9	–	–	22	8
HPAT	2	5	3	5.7	2	5.4	6	5.6	2	5.9	–	–	15	5.5
HSEM	7	17.5	18	34	2	5.4	5	4.6	1	2.9	–	–	33	12
HSYN	16	40	13	24.5	6	16.2	4	3.7	1	2.9	3	100	43	15.6
TYPO	7	17.5	4	7.5	5	13.5	13	12	8	23.5	–	–	37	13.5
Lesson Total	40	100	53	100	37	100	108	100	34	100	3	100	275	100

[a] One student did not submit the Lesson 5 assignment Web page. [b] The Lesson 6 assignment was to resubmit prior Web pages with all revisions and repairs completed. Only one student made new HTML syntax errors (HSYN, *n*=3) which warranted a new feedback document (formative assessment), because the errors prevented the Web pages from displaying properly in an Internet browser.

Reasons for student misunderstandings. The student questions recorded in the observation data (see Table 8) and the errors collected during document analysis of students' assignments (see Table 13) were then compiled by misunderstanding code for analysis (see Table 14). For the unit *(N=523)*, the most frequent student misunderstanding/error related to HTML syntax (HSYN, *n*=68), while CSS content issues (CCON, *n*=13) were least frequent (see Figure 9). Again, misunderstandings were shared by students and present in assignments regardless of prior experience.

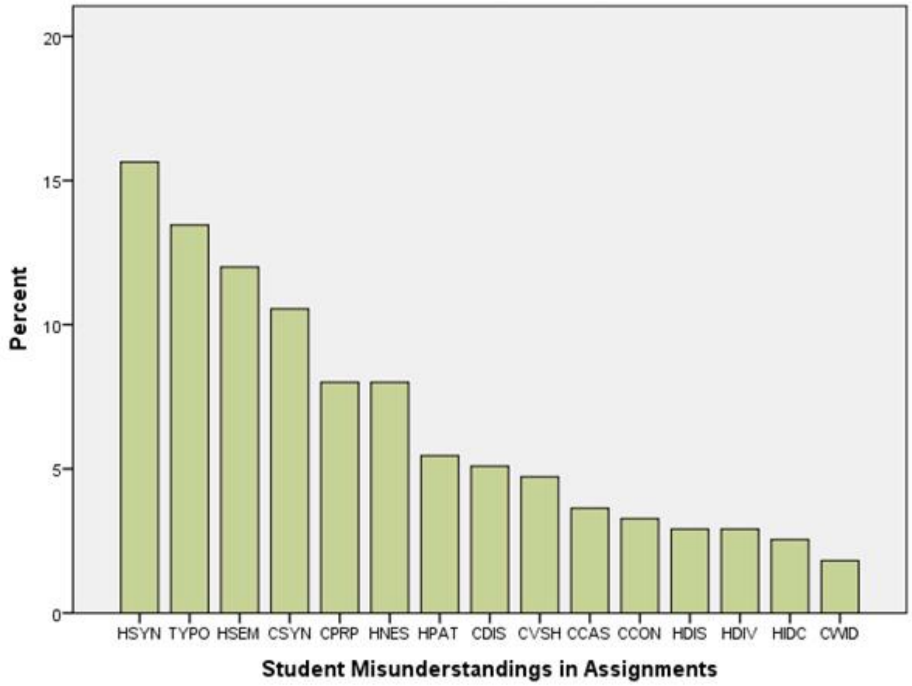

Figure 8. Student misunderstandings in homework assignments (unit totals by percent).

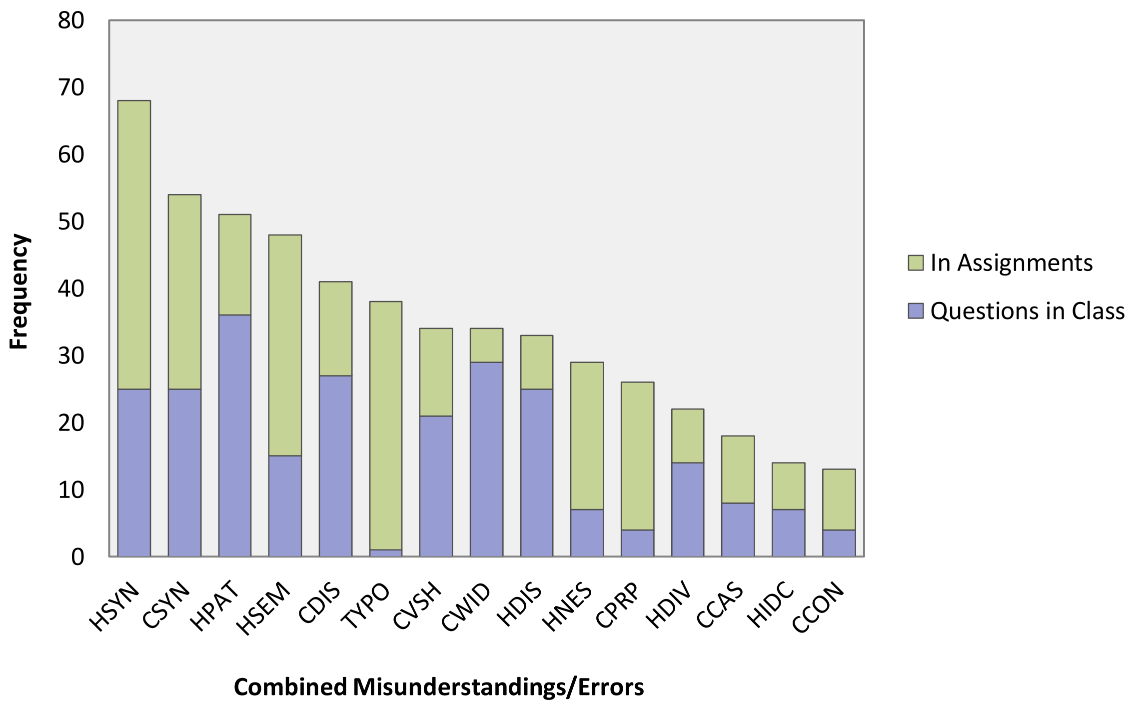

Figure 9. Combined student misunderstandings/errors from in-class questions and homework assignments (unit totals by frequency).

Analysis of the combined student question and document analysis data, coupled with further review of the observation data and lesson plans themselves, revealed the following six primary reasons for student misunderstandings during curriculum unit implementation:

1. Insufficient experience-building activities during class-time (e.g., the quantity of new information in Lessons 1 and 2 was overwhelming to novices; more practice was needed instead).
2. Insufficient procedural introduction to abstract HTML concepts, like adding tags semantically (e.g., misunderstandings on HTML basics persisted even after the unit transitioned to CSS).
3. Insufficient demonstration of how code translates to display, especially CSS styles (e.g., not enough HTML editor/Internet browser split-screen during demonstration).
4. Insufficient procedural instructions for abstract CSS layout concepts, like positioning and flow (e.g., layout and spacing concepts were the most difficult for students to grasp).
5. Insufficient establishment of the project website content and information architecture (e.g., lack of content led to a hesitation to experiment with layout in the homework assignments).
6. Insufficient introduction to computer literacy concepts, especially file pathing (e.g., the instructor had to model file management after student questions revealed misunderstandings on computer functionality).

Table 14

Combined Misunderstandings/Errors (N=523) from Student Questions and Assignments

	Questions in Class		In Assignments		Unit Total	
Errors	*n*	%	*n*	%	*n*	%
CCAS	8	3.2	10	3.6	18	3.4
CCON	4	1.6	9	3.3	13	2.5
CDIS	27	10.9	14	5.1	41	7.8
CPRP	4	1.6	22	8	26	5
CSYN	25	10.1	29	10.5	54	10.3
CVSH	21	8.5	13	4.7	34	6.5
CWID	29	11.7	5	1.8	34	6.5
HDIS	25	10.1	8	2.9	33	6.3
HDIV	14	5.6	8	2.9	22	4.2
HIDC	7	2.8	7	2.5	14	2.7
HNES	7	2.8	22	8	29	5.5
HPAT	36	14.5	15	5.5	51	9.8
HSEM	15	6	33	12	48	9.2
HSYN	25	10.1	43	15.6	68	13
TYPO	1	0.4	37	13.5	38	7.3
Total	248	100	275	100	523	100

Summary

The results of the data collection and analysis as they relate to each aspect of the research questions are presented in this chapter. The observation data revealed the trends in and reasons for deviation from the lesson plans during implementation. Questions from students (34.1%) and the strategic expansion of coverage by the instructor (25%) were the most frequent reasons for deviation. Student questions and misunderstandings during class were also collected from the video observations, as were the types of technical difficulties encountered with computer software and hardware (since these impacted students' understanding). The document analysis data revealed the student misunderstandings/errors present in their assignments. Analysis of the assignment errors combined with the data on student questions asked in class revealed the trends in and reasons for persistent misunderstandings: participants struggled with HTML and CSS computer language syntax as well as computer literacy and Web design concepts, as the lesson plans failed to sufficiently meet the needs of novice students. Deductive qualitative analysis and a priori reasoning were used, since the study was a continuation of an investigation begun during piloting.

The next chapter presents a discussion of the results reported here and the conclusions drawn from them regarding concept-based Web design instruction, as well as limitations, and implications for reformulating the curriculum unit specifically and Web design education generally.

Chapter Five

Discussion

"Teachers, especially at the high school and college level, often fail to adequately consider the deficiencies in the students' prior experiences*—and then wrongly think that what they need is more* knowledge.*" (Wiggins & McTighe, 2005, p. 208)*

As engagement with the Internet and Web technology continues to permeate daily life, there is often the educational assumption that college students can teach themselves to build or manipulate Web pages (Kalman & Ellis, 2007; Karper, 2004; Wiggins & McTighe, 2005). An industry of Web design textbooks and software resources has perpetuated assumptions of self-instruction ease (Bluttman & Cottrell, 2012; Tuck, 2011), and as a result there has been a dearth of research-based exploration of how to effectively teach Web design (Connolly, 2012; Karper, 2004; Kotamraju, 1999; Park et al., 2013) and details of the difficulties involved. Most Web design pedagogy is simply governed by an overreliance on textbooks that focus on procedural knowledge and inauthentic assessment rather than learner-centered conceptual understanding (Wiggins & McTighe, 2005)—the textbook-model or software-model of instruction was established in Web design's early history and the instructional strategies endorsed by modern learning theories have not been consistently applied to the subject since. The rapid technological change associated with Web design's complex (and still-evolving) combination of computer, visual and information literacies has also served as a deterrent to detailed investigation (Karper, 2004; Kotamraju, 1999; Victor, 2002).

In an effort to contribute to the knowledge-base on Web design instruction, this study evaluated a six-lesson curriculum unit that was developed to teach XHTML, Cascading Style Sheets (CSS) and Web design conceptually, not just procedurally. The curriculum unit used as its framework the ‘new science of learning’ (Bransford et al., 2000—see Figure 3 in chapter 2) and integrated instructional strategies determined effective by behaviorism, cognitivism and constructivism (i.e., learning theories), strategies such as student-controlled learning, scaffolding and individualized assignments (Kalman & Ellis, 2007). An authentic implementation of the curriculum unit was conducted, during which observation data and student documents were collected and then analyzed deductively using piloting data as a starting point. The study aimed to document the persistent misunderstandings encountered by novices and non-technical learners, particularly for the underrepresented topic of CSS, because Web design instructors “need to know where first-time designers may encounter problems or need support that is different from what is currently being provided” (Karper, 2004, p. 26-27). Particular attention was dedicated to identifying not just what should be taught in Web design instruction but how it should be taught.

To address its framing questions, this study found five reasons for deviation from the lesson plans and six primary reasons for persistent student misunderstandings during curriculum unit implementation; these reasons reflect insufficiencies in the curriculum unit itself. The conclusions drawn from the study’s findings are discussed in this chapter, as are the study’s implications for concept-based Web design education. Limitations to the study and recommendations for future research are also addressed.

Review of the Findings

As reported in chapter 4, five reasons for deviation from the lesson plans were identified: Student Questions, Expansion/Strategy, Class Ended, Instructor Error and Lesson Plan Defect (see Figure 5; see Table 6 for details). Student questions instigating the most deviations (STUQ=34.1%) can be expected in learner-centered, constructivist instruction, where student initiative is advocated and emphasis is placed on students taking control of their own learning (Bransford et al., 2000; Smaldino et al., 2005). But, the misunderstandings documented and the instructor's strategic attempts (EXPS=25%) to alleviate and address student confusion indicate six findings about the curriculum unit:

1. Insufficient experience-building activities during class-time (e.g., the quantity of new information in Lessons 1 and 2 was overwhelming to novices; more practice was needed instead).
2. Insufficient procedural introduction to abstract HTML concepts, like adding tags semantically (e.g., misunderstandings on HTML basics persisted even after the unit transitioned to CSS).
3. Insufficient demonstration of how code translates to display, especially CSS styles (e.g., not enough HTML editor/Internet browser split-screen during demonstration).
4. Insufficient procedural instructions for abstract CSS layout concepts, like positioning and flow (e.g., layout and spacing concepts were the most difficult for students to grasp).

5. Insufficient establishment of the project website content and information architecture (e.g., lack of content led to a hesitation to experiment with layout in the homework assignments).
6. Insufficient introduction to computer literacy concepts, especially file pathing (e.g., the instructor had to model file management after student questions revealed misunderstandings of basic computer functionality).

These findings revealed insufficiencies in both how the Web design or computer literacy content was introduced and in the unit's application of instructional strategies—the need for more in-class practice and more demonstration/modeling being indicated at multiple points. And, though the unit's introduction to HTML and CSS had been deemed rudimental by the subject matter experts consulted during curriculum unit development, as well as the study's Expert Review Panel, the data revealed that Lessons 1 and 2 in particular were still not simplified enough, not concrete or procedural enough, for novices. In sum: too much information was presented at the start, covered too quickly, and knowledge was detrimentally prioritized over experience.

The surplus of information in the lesson plans was also evidenced by class ended deviations (ENDC=15.9%) occurring in five of the six lessons. The routine extension of review, discussion, and demonstration sections to accommodate student questions or the instructor's expanded explanations resulted in practice sections being truncated, simply because they were the last section in the lesson plans—this in turn contributed to the lack of sufficient in-class experience-building. This study ultimately supported Jakovljevic et al.'s (2004) conclusion that "the lack of sufficient time impacted on the quality of the teaching and on the learners' design solutions" (p. 280). (Note: though the lesson plans were developed and piloted for a 2.5 hour class duration, the study site turned out not to afford this length of time—see Study Limitations. The misunderstandings documented during the study had persisted throughout pilotings with full durations, regardless.)

Also, three lessons contained minor instructor error deviations (INST=13.6%). Most noteworthy: four of the six instructor errors occurred in Lesson 1. Just as the early lesson plans presented "too many disconnected facts in too short a time—the 'mile wide, inch deep' problem" (Bransford et al., 2000, p. 24) to students, they also seemed to present too much information to the instructor. The quantity of detail that the instructor was expected to follow led to increased chance of accidentally skipping demonstration steps or providing incomplete coverage or poor paraphrasing, for example.

Deviations also occurred in three lessons because of minimal defects (LESS=11.4%) or mechanical flaws in the lesson plans in the form of unclear instructions and sections that needed re-sequencing or merging. For example, the link pathing review and assignment demonstration were separated in Lesson 2, which proved artificial based on student questions and led to instructor deviation (re-sequencing). Revisions should of course be made to the curriculum unit to eliminate these defects.

Conclusions

This study's results were primarily a replication and extension of prior conclusions in the literature regarding Web design instruction, including Dick's (2006) assertion that "the decision to teach HTML coding as an alternative to using packaged Web-design software has shown mixed results" (p. 206). It was also an affirmation of educational findings on the importance of strategies such as experience-building, demonstration and scaffolding for overcoming novice's lack of prior knowledge. Unique to this study however, are its details on the persistent misunderstandings encountered by non-technical learners and the outcomes from introducing Web design conceptually and procedurally simultaneously. The study also demonstrated how novices' learning needs are sometimes in conflict with calls for authenticity: "meaningful learning occurs within the environment that resembles the real world, which is often a very *complex environment*. Finding such a complex environment that is conducive to the needs of the learner is often difficult" (Newby et al., 2006, p. 37). Employing authentic, constructivist activities during Web design instruction for example, increases the complexity of the course (Park & Wiedenbeck, 2011).

Experience before knowledge. The new science of learning concludes that "what [students] will learn depends on how much they know already" (Bransford et al., 2000, p. 50) and since the study participants were non-technical learners with no or minimal prior experience with computer languages on which to build, they struggled considerably with the novelty of HTML (see Table 2 in chapter 3 for prior experiences)—"If you have never designed a Web page using HTML coding, you probably find this language quite complicated with the use of the less than (<) and greater than (>) signs to begin and end all instructions or tags" (Niess et al., 2008, p. 190). As a result, Lessons 1 and 2 were the most problematic for students and underwent the most deviation during implementation (see Figure 4). The curriculum unit's attempts to introduce HTML conceptually by relating it to students' prior Internet and computer experiences proved ineffective, because students were otherwise overwhelmed by the quantity and complexity of new information. The study findings supported existing educational conclusions that instruction should "postpone the teaching of definitions, rules, and theories until they are needed to make sense of experience" (Wiggins & McTighe, 2005, p. 220)—a crucial constructivist strategy for addressing novices' lack of prior knowledge. All six lesson plans did emphasize experience—they contained practice sections, plus sections where students could type along with demonstration that models the homework, followed by homework assignments that continued the practice begun in class. But, Wiggins and McTighe's (2005) argument that "teachers, especially at the high school and college level, often fail to adequately consider the deficiencies in the students' prior *experiences*—and then wrongly think that what they need is more *knowledge*" (p. 208) was still perpetuated.

The study also demonstrated the primacy of novices' "need to take time to explore underlying concepts and to generate connections to other information they possess" (Bransford et al., 2000, p. 58), as evidenced by the endurance throughout the implementation of questions and errors on HTML basics like syntax, semantics and nesting—topics introduced in Lesson 1—and particularly the reemergence of these misunderstandings in Lesson 6's cumulative review session. HTML syntax was the most common area of student misunderstanding during the study (HSYN=13%—see Figure 9), a result which mirrored the findings of Park and Wiedenbeck (2011) and Blackwell (2002) on students' difficulty with HTML basics. Park and Wiedenbeck's (2011) conclusion was also that students need sufficient time:

> [our] findings suggest that instead of a web development course that "sprints" toward programming, a more elementary version that delves deeply into HTML and CSS…may better serve some learners. Particularly for students without prior

> experience, a few weeks of instruction may not be a sufficient introduction. (p. 131)

The students' persistent misunderstanding of HTML can also be seen to support the argument that HTML, not just CSS, "has too steep of a learning curve for the majority of users" (Notess, 2006, p. 45; see also Kotamraju, 1999).

The novices' memory was also likely overloaded by the initial quantity of new information, a known difficulty in teaching coding and computer languages: "a large number of students achieve only low grades and become disillusioned with the subject [computer programming]…one of the reasons for the above is that students experience very high cognitive load during their learning" (Garner, 2002, p. 578). Information processing strategies for reducing cognitive overload, such as discussion connecting new information to prior experiences, concept anchoring, demonstration of worked-out examples, and content chunking (Newby et al., 2006; Schnotz & Kurschner, 2007), had been integrated in the unit of course, but again, there was simply too much previously un-encountered information presented for these strategies to be as effective as desired—again, more practice (and more time) was needed by the participants in order for them to develop the mental connections and models (i.e., schemata) they needed to make sense of the new Web design information (Newby et al., 2006).

The third-most deviated lesson, Lesson 5, also spoke to the novices' need for additional experience-building. After making 37 errors total on Lesson 3's homework assignment, students made 108 errors on the Lesson 4 assignment (see Table 13). Lesson 5 was then the class where these 108 errors were reviewed and discussed, and the instructor strategically deviated from the lesson plan by converting one hour of the review session into an ‘intervention’, a practice session where students troubleshot and revised their assignments with classmates’ and the instructor’s assistance. (The curriculum unit front matter contained notes for such scenarios.) The tripling of errors on assignment 4 may in part be attributed to the complexity of the abstract CSS layout concepts introduced in Lesson 4 and to the fact that students were now working in-depth in not one but two computer languages (XHTML and CSS), but the data also implied that students were once again overwhelmed with (more) new information and not afforded the adequate time or practice needed for connecting the new CSS concepts to previous HTML concepts. For example, students struggled to effectively add DIV tags to their Lesson 4 assignment Web pages as part of the process for creating layout columns, even though they had effectively added HTML tags for three previous lessons—student questions on CSS layout (CWID) then ranked highest during the Lesson 5 class while questions on the related concepts of HTML DIV usage (HDIV) and how CSS code translates to display (CDIS) were both ranked second (see Table 8).

Participants reacted favorably to the added in-class practice, saying during the Lesson 5 class, and especially in Lesson 6’s curriculum unit debriefing, that “our time together is more helpful than anything…the quick feedback definitely helps” (Student 1). This was a replication of Karper’s (2004) finding that “participants often focused on their need for direct instruction from others in order to help them learn, or their need to have a resource person available in case they get stuck” (p. 85), as well as Stepp et al.’s (2009) findings that lab sessions where instructors can quickly help students locate and fix unseen coding errors are crucial to effective learning. Considering this, these studies provide a rationale for the prioritization of scaffolding strategies during Web design instruction over self-instruction strategies, perhaps:

> A scaffold is a support that a teacher or learning environment provides to a learner to assist him or her in a range of cognitive tasks, from the understanding of a task and mastering of a skill to the solving of a problem. Scaffolding is an important feature of Vygotsky's (1962) social development theory… this theory holds [that learning] requires social interaction through expert guidance and peer collaboration. (Doering & Veletsianos, 2007, p. 109)

Procedure before concept. Study participants struggled to conceptualize HTML's function (markup) and its relationship to content (text, images) and design, along with struggling to code it. Difficulty may partly be ascribed to the quantity or complexity of HTML concepts which, as discussed, overwhelmed the novices initially, but the types of questions/errors documented also indicated insufficient amounts of procedural or step-by-step instructions on how/when to add HTML tags. (This was most evident in their homework assignments, where HTML syntax (HSYN) errors persisted in all six lessons (see Table 13) and as mentioned, Lesson 4's introduction of DIV tags caused a return to confusion on how to add HTML tags, for example.) The implication was also that expanded procedural introductions should occur before abstract conceptual information was introduced, or that by combining conceptual information with procedural information, the early lessons were not sufficiently concrete for novices (see Implications for Teaching Web Design Conceptually). The demonstration portions of each lesson plan did employ the behaviorist strategy of direct instruction (Lever-Duffy et al., 2005) and were intended to address novices' need for procedural instruction by modeling coding—"students approaching a subject new to them learn best from structured presentations even if they have a learning style that would otherwise indicate more open-ended, unstructured methods" (Smaldino et al., 2005, p. 50). The study's finding that more demonstration was needed support this claim.

Review of the data revealed that the curriculum unit's conceptual introduction to CSS layout, like the introduction to HTML discussed, also lacked a sufficiently procedural focus needed by novices: "Students without practice in graphic design need some guidelines to conceive the visual expression of Web pages; otherwise, they cannot begin to design it at all, or make the visual design heedlessly" (Ariga & Watanabe, 2008, p. 827). Recall though that visual design is "an abstract process [that] cannot be reduced to sequential procedures or lists of a guideline for adequate design" (Ariga & Watanabe, 2008, p. 817)—this is what Lesson 4's introduction to CSS layout likely reflected. Designing a Web page is "substantially different from composing processes in other media" (Karper, 2004, p. 107) and there is no set procedural model for CSS layout, because the process can differ for every Web page. The unit's CSS lessons contained general design guidelines and modeling of procedures for adding layout columns, though—e.g., Lesson 4 contained an in-class activity for sketching layout columns followed by demonstration of coding to create these columns—but the persistence of student questions still emphasized their deficiency: particularly abstract concepts like tag flow—the order in which tags are typed in the code may determine the display and position of content on the Web page—could have been presented even more concretely, more procedurally, so that the reemergence of student questions in Lesson 6 on how the order of HTML tags works in conjunction with CSS properties to create layout (CVSH—

see Table 8), for example, could be diminished. Participants' difficulty with CSS layout again reflected in part their lack of prior experience on which to build—just as these novices had no or minimal experience with computer languages, they also had no or minimal visual design experience (see Table 2).

Demonstration before discussion. In keeping with the conclusion that a procedural introduction to HTML or CSS should be attempted before being augmented with conceptual information, the study also demonstrated the primacy of novices' need for exact demonstration of how HTML and CSS code translates to display. (The study replicated computer literacy conclusions that technology novices prefer demonstration and lecture when encountering novel information (Kalman & Ellis, 2007).) The participants struggled to recognize how adding an HTML tag also adds a default presentation style, for example, and that displayed styles can be altered, added or removed using CSS. How non-visual computer code translates into visual display is perhaps the most difficult concept in Web design for novices to grasp (Andrew & Yank, 2008), because

> what the user sees when authoring the page is not necessarily what he or she gets when viewing it – a conflict with the ideal of WYSIWYG. What he or she is manipulating is not a concrete instance of the desired result, but an abstract notation defining required behavior in different circumstances – a conflict with the ideal of direct manipulation. (Blackwell, 2002, p. 3)

In other words, students learning to code computer languages are actually learning to create 'representational notations' (Blackwell, 2002) which then display as a visual object; this is much more difficult than if they could directly manipulate the object itself.

Questions in class then about how CSS translates to display were the third most frequently asked (CDIS—see Figure 6 and Table 8), with questions about how HTML translates to display (HDIS) fifth most frequently asked. How CSS translates to display was also fifth-highest in combined questions and errors (see Figure 9).

The participants suggested their own solution to this issue by requesting Web editor (code) and Internet browser (display preview) software split-screen during demonstrations: "so, is there a way to have…do a split…like, I use HTML and design split?" (Student 5, Lesson 6). The importance of split-screen during learning Web design is supported by Park et al.'s (2013) finding that "at all levels, feedback provided by the web editor's live preview panel was instrumental in detecting and resolving errors. As participants typed their code, they were able to immediately test it as the page rendered in real time" (p. 82). The implication was also that demonstration of display should be presented before in-depth discussion or lecture on how to achieve the displayed styles began. Review of the observation videos confirmed that demonstration sections during the implementation contained both coding and display, but as indicated, the novices deemed it inadequate. Review of the lesson plans themselves then revealed that the demonstration pattern outlined was to first compose the code and then discuss display view or browser preview later, rather than side-by-side display or switching to design view after typing each line of code. This was likely a function of making the HTML and CSS code examples in the lesson plans easier for the instructor to read and type, but the lesson plans also did not emphasize a need for side-by-side display or systematic, line-by-line demonstration. This was a likely example of the detriment of an Expert Blind Spot (Wiggins & McTighe, 2005), in which experts' "conceptual understanding allows them to extract a level of meaning from information that is not apparent to novices" (Bransford et al., 2000, p. 17). In other words: an instructor may type five lines of CSS properties and visualize how they will display together to style text, while novices instead need to concretely witness how each property/line of code displays as it is added.

Student misunderstandings on how code translates to display also persisted simply because of the procedural (and conceptual) complexity CSS uses to create displayed styles (Andrew & Yank, 2008; Powell, 2010). In combined misunderstandings, CSS syntax was second-highest behind only HTML syntax (see Figure 9), representing students' difficulty not just with the CSS language's quantity of variables, but the quantity of ways in which those variables may be combined to create styles. (Recall that HTML contains a finite number of tags compared to CSS, which has exponentially more properties and values (Lie & Saarela, 1999; Teague, 2008).) In total, CSS basics like syntax (CSYN), properties (CPRP), and cascading (CCAS) represented fewer student misunderstandings during the implementation than the three equivalent, foundational HTML categories (HSYN, HSEM, HNES—see Table 14), but the CSS misunderstandings still provide a rationale for expanded demonstration sections that more effectively explicate the relationship of code to display.

Students' difficulty conceptualizing how code translates to display also contributed to their misunderstandings of how CSS is used to create Web page layout. Questions in class on CSS layout ranked second (CWID—see Figure 6) behind only HTML pathing, and when grouped, design concepts like CSS layout, CSS display and HTML display represented the second largest grouping of misunderstandings/errors behind coding mechanics categories (e.g. HTML syntax (HSYN), CSS syntax (CSYN)—

see Table 14. (Note: the curriculum unit's known limitation of prioritizing instruction on coding over instruction on design explains in part why coding misunderstandings were the more prevalent of the two—see Study Limitations.)

During Lesson 6's curriculum unit debriefing, for example, students were asked "Where do you feel that you had the most difficulty learning HTML/CSS?" They answered that CSS layout and positioning were the areas of most concern: "it was this last section of the CSS formatting…making things properly space—because I don't have an intuitive sense in terms of sizes of ems and pixels and percentages and trying to understand how big it'll be when I block it out…" (Student 1, Lesson 6). (Lesson 4 was the 'last section'. It introduced CSS layout and how to create columns using width, float and box model CSS properties.) This again supports claims of the importance of demonstration strategies for novices' learning (Kalman & Ellis, 2007; Smaldino et al., 2005), technological or otherwise, as well as adding to the study's conclusion that experience-building is the priority for effectively learning Web design.

Content before practice. Students' misunderstanding of CSS layout concepts led to a hesitation to experiment with layout and design in the homework assignments, as evidenced by student questions in class on CSS layout ranking second (CWID—see Figure 6), while CSS layout errors in the assignments ranked last (see Figure 8). This low frequency was not an indication that students could execute CSS layout effectively, only struggling with it conceptually. Instead, the document analysis indicated that students made fewer CSS layout errors in assignments, because they mimicked the demonstration code more closely than in earlier assignments or typed only the minimal CSS layout properties called for by the assignments—there was less experimentation and therefore less engagement with CSS layout concepts. The observation data then revealed a lack of sufficient content around which to create layout as a determining factor: each of the participants chose to build a Web site related to another school or work project—the curriculum unit encourages using the constructivist strategy of an authentic Web site project, so that students will be “highly motivated by the activity because they gain a sense of ownership in the product and in their learning” (Lim et al., 2003, p. 14; see also Kalman & Ellis, 2007). As the curriculum unit implementation progressed however, it emerged that most of the content for these outside projects did not yet exist. (Not coincidentally, the student with the most available text, links and images experimented the most in their CSS layout and styles.) The instructor then frequently had to prompt students to brainstorm additional content or use filler text (i.e., lorem ipsum) with which to complete their later homework assignments. The conclusion was then that the curriculum unit lacked sufficient emphasis on requiring composing or collecting of content, and that content must be finalized before students can effectively practice more advanced CSS. Requiring content development early in the unit may also increase students’ experience with their area of most difficulty in the assignments, HTML syntax (discussed earlier)—i.e., more content necessitates practicing with more HTML tags.

Participants’ uncertainty over their Web site’s final content also led to their hesitation to practice Web site organization and information architecture concepts in the assignments, as evidenced by CSS content (CCON) ranking last in combined questions/ errors and HTML ID and class naming errors ranking second-last (HIDC—see Figure 9)—both of these topics concern the relationship of code and content, among other things. Again, such low instances did not indicate improved understanding of these two topics. Instead, without abundant content to organize, Lesson 3's activity on adding navigational menus remained too abstract for the novices, and this in turn led to hesitation to practice CSS styling of menus in their Lesson 4 assignment.

The lack of established content also called into question the effectiveness of the Web site template assignment in Lesson 5: the document analysis revealed that because they lacked finalized content, students made minimal revision to the template code. The aim of the Lesson 5 assignment was to offer students the experience of working with HTML and CSS written by someone else—another authentic task, since Web designers rarely interact only with code they compose—and to gauge how well they could transfer what they had learned creating their own code to HTML and CSS code in general. (The template assignment aligned with the new science of learning's conclusions that students should be better prepared to transfer what they have learned to new problems and settings (Bransford et al., 2000—see Figure 3).) But, again, all but one of the students made only minimal alteration to the existing Web site template code—this diminished students' experience-building as well as made assessing how well they could transfer their understanding of topics and concepts in the lesson plans more difficult to document. The participants were able to adequately complete the template assignments (e.g., see Figures 10 and 11) and interact with template code, but their engagement and practice had simply been lessened by the lack of content. That the curriculum unit did not articulate the need to enforce content establishment activities or submission became detrimental to both students' learning and application.

Figure 10. Student 4’s finalized Web site project using a template.

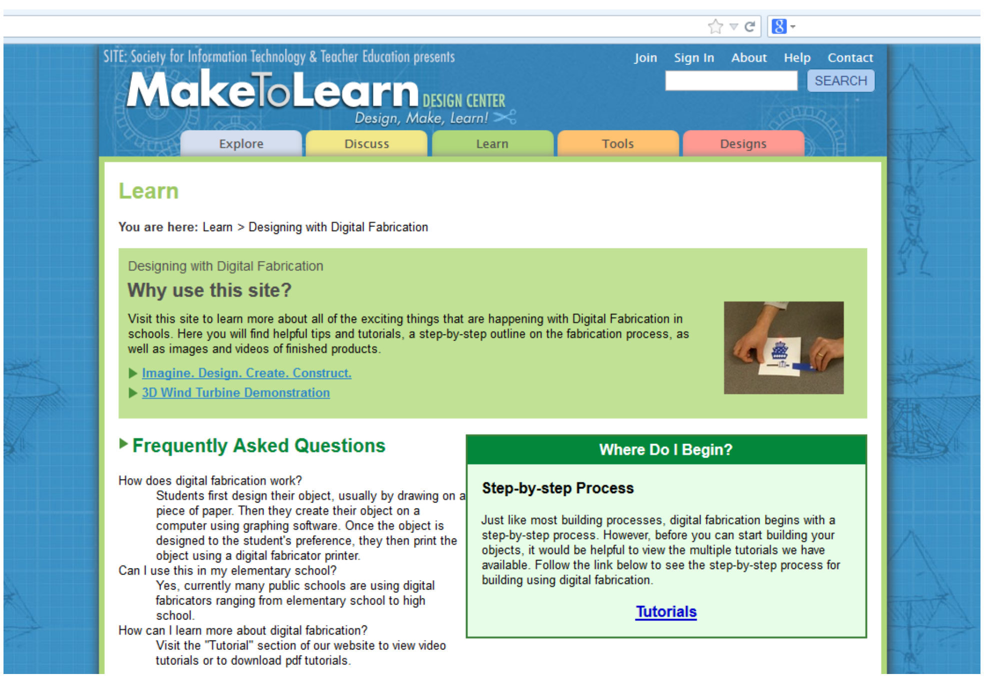

Figure 11. Student 5’s finalized Web site project using a template.

Computer literacy before Internet literacy. Student misunderstandings of computer literacy concepts such as file pathing, file management and Web servers also persisted throughout the curriculum unit implementation. (File pathing refers to the exact location, the address, of a file.) Though the participants operated their computers daily for a myriad of college or professional activities (see Table 2), the study revealed only minimal prior consideration of the nature of computer functionality. The participants lacked an effective 'model of computers' on which to build their conceptual understanding, and this lack of a model has been found to be "a serious obstacle to learning" computer topics (Ben-Ari, 1998, p. 259; see also Park & Wiedenbeck, 2011). Participants had never conceptualized how computer folders represent a physical location on a hard drive, for example, nor had they conceptualized that clicking on a hyperlink on a Web page retrieves/displays a file from an exact location: "creating hyperlinks for Web pages is fundamentally different…the concept proved to be quite overwhelming for the students who were unfamiliar with the process of composing Web pages" (Dick, 2006, p. 211). This study supported Dick's conclusions, as well as Park and Wiedenbeck's (2011) findings on the persistence of HTML pathing errors throughout Web development instruction: HTML pathing was the most prevalent student question category observed during the implementation (HPAT=14.5%—see Figure 6); HTML pathing also ranked third-highest in total combined misunderstandings behind only the two coding mechanics categories (HSYN, CSYN—see Figure 9). (For research specifically on the mistakes Web development students make when file pathing and linking, see Miller, Perković and Settle (2010).)

The implication was also that introductions to computer literacy should be sequenced before Internet literacy or Web design topics, a conclusion also reached by Ben-Ari (1998) for computer science majors: “programming exercises should be delayed until class discussion has enabled the construction of a good model of the computer” (p. 260). For example, Lesson 1 contained a “What is a Web page?” section in which Web page basics (functionality, code vs. display) were discussed conceptually as part of the ‘big picture’ of the unit. The novices’ questions revealed that before they could develop their model of a Web page, so to speak, they needed an even more general introduction to computers, how computer operating systems manage files, how Web browsers are a form of software and how file types correspond to different software—an HTML file and a JPG photo will both display in a Web browser, for example, but browsers are not actually an editing software. (There are exceptions to this idea with online tools like Google Docs or Pixlr, as well as browser extensions for manipulating Web pages.) It should be noted that students’ questions during the “What is a Web page?” section likely also stemmed from it being conceptual rather than procedural, and as discussed above, the novices struggled with the conceptual areas in the early lessons.

Review of the lesson plans revealed that though the curriculum unit included multiple activities and demonstration devoted to HTML pathing and linking to Web pages, images, etc., as well as repeated procedural directions in each assignment for saving files to the Web server (i.e., submitting), introductions to computer literacy topics were cursory in favor of focusing on Web design topics like effective HTML/CSS coding or Internet literacy strategies like self-instruction. Student questions then prompted the instructor to deviate from the lesson plans by modeling computer literacy during review sessions or practice sections, reiterating in Lesson 4, for example, how online file storage like Dropbox can host a Web site but only when the /public/ folder is used, that there is a minor conceptual difference between online storage and a Web server.

File management was also the most frequent area of technical difficulty (FILE=46.2%—see Figure 7) encountered during the study, followed distantly by Web server difficulty (SERV=17.9%). Managing multiple HTML, CSS and media files, plus resubmitting assignment files with corrections, simply proved a confusing scenario for the novices. They struggled to retrieve and organize their files, to differentiate one Web page from another, and to link/path these files together. Instructions in the lesson plans for establishing file naming conventions (e.g., student's last name + assignment number) and repeated direction by the instructor to "save all your files in the same folder so you can find them" were ineffective or unheeded by students. The need to save copies of Web pages—to the class folder or even just for making revisions—was especially problematic. The study ultimately confirmed Maddux et al.'s (2008) conclusions that the skills to upload/download files—let alone knowledge of general Web page principles—are missing from many college students' basic computer literacy knowledge and experience.

Student feedback in Lesson 6's curriculum unit debriefing eventually suggested that an in-class, experience-building computer literacy activity was warranted—a "here's a file structure for each week" (Student 1, Lesson 6) activity, e.g.,—so that students can practice managing and pathing to files with in-class modeling and scaffolding. This suggestion is supported by Ben-Ari's (1998) recommendation for computer literacy instruction: "group assignments and [in-class] labs should be preferable to individual homework exercises, because they soften the brutality of the interaction with the computer and facilitate the social interaction that is apparently necessary for successful construction" (p. 261). (Note: This computer literacy activity should perhaps be located before the unit in a 'lesson 0', since as discussed, the study's conclusion is that Lesson 1 already contains a surplus of new information for novices.)

The implication was also that an even more rudimentary pattern for saving/ submitting files should be employed—students working only in the online class folder and not on their local computers, perhaps, with the conceptual introduction to Web servers postponed as late as Lesson 5, where it could be paired with the introduction to templates. This conclusion would reduce the authenticity of students' experiences posting Web pages—constructivism stresses the importance of authentic problems for effective learning, after all (Lever-Duffy et al., 2005; Lim et al., 2003; Smaldino et al., 2005)—but limiting the process or procedure may at least alleviate some persistence or reemergence of file pathing and management misunderstandings. (HTML pathing was the most prevalent student question category during Lessons 1-3, and though lessened, questions reemerged in Lessons 5 and 6—see Table 8. Again, this was a replication of Park and Wiedenbeck's (2011) results on the persistence of HTML pathing issues.) The lesson plans' employment of authentic posting procedures is likely another example of an Expert Blind Spot (Wiggins & McTighe, 2005): expert Web designers interact with multiple copies of a Web site and post files to Web servers after each revision, but again, this scenario was too complex for the study's novices and a simplified introduction to computer literacy concepts was instead warranted.

Implications for Teaching Web Design Conceptually

As discussed throughout the conclusions, the study participants frequently 'pushed back' against the abstractions and complexity of Web design concepts by requesting more procedural instruction. The study even replicated Karper's (2004) finding that novices

> attempted to reduce all technological aspects of Web design…to a series of precise context-specific steps which they could then repeat over and over again, rather than approaching technology use in this situation as a series of flexible concepts and procedures that could be transferred to different situations. (p. 133)

This reflected the students' comfort and prior experiences with procedural learning (Jakovljevic et al., 2004; Karper, 2004), as well as the fact that during learning, "the tendency is to use the same strategy or approach used the last time, thus diminishing the cognitive challenge of the problem" (Raths, 2002, p. 236-237). But, the conclusion should also be that the curriculum unit's application of conceptual learning strategies could be more effective. The lesson plans' combination of concepts and procedures did not effectively ease students into the uniqueness of Web design with "direct, concrete kinds of experiences" (Smaldino et al., 2005, p. 49), for example. As discussed, introducing HTML both conceptually and procedurally in Lesson 1 likely contributed to the participants being overwhelmed by new information (see Experience before knowledge section). The curriculum unit instead could have begun with procedural learning alone and then transitioned or 'ramped up' to conceptual learning of thinking skills and decision-making—an issue of experience-building for novices, yet again. The lesson plans could also better strategize for modeling/demonstration of conceptual understanding by the instructor (of computer literacy topics like file management, for example—see Computer literacy before Internet literacy). Again, recall that demonstration is a crucial conceptual learning strategy recommended by Royal (2005). Many of the revisions suggested by the study results would likely address the conceptual learning difficulties encountered during the implementation. With expanded demonstration by the instructor on abstract concepts like HTML semantics, for example, persistent student misunderstandings of basic concepts may be targeted.

To conclude from the study findings that Web design should be introduced only procedurally to novices, not conceptually, would be counter to the recommendations of not only Web design educators but also modern learning theories including constructivism (Lever-Duffy et al., 2005) and the new science of learning (Bransford et al, 2000)—"a balanced combination of theory and practice is essential in studying web design. Only those who manage to combine these two things can become good web designers" (Krunić et al., 2006, p. 326; see also Jakovljevic et al., 2004). Recall that the new science of learning argues that education "is better conceived as helping students develop the intellectual tools and learning strategies needed to acquire the knowledge that allows people to think productively" (Bransford et al., 2000, p. 5). This argument is particularly relevant for Web design, because of the rapid change inherent to the topic: Web technology evolves quickly and frequently—the procedures, software and hardware are in a state of constant change—and more than anything Web design students need to learn the concepts that allow them to adapt to changes (Kalman & Ellis, 2007). This study demonstrated at least what a complex scenario learning the transferable concepts/ reasons behind the procedure, while also learning the procedures themselves, can be for novices.

The data did reveal that by implementation's end, students were more receptive to conceptual learning, requesting additional conceptual instruction on CSS layout and design, for example: "because I don't have any background in…design…I wanted to know more 'here are ways to break up your space', and you know, general color schemes…[how] to draw on my experience with photography in terms of breaking [CSS layout] up" (Student 1, Lesson 6). Also, the participants responded favorably to the curriculum unit's conspicuous references to strategies (Rockwell, 2008), including sanctioning of self-instructional strategies, which were typically conceptual rather than procedural. They indicated in Lesson 6's curriculum unit debriefing, for example, that the experience gained interacting with the Web-based instructional materials increased their willingness to seek out information online, to self-instruct:

> When talking about educators…you hear "be careful what you find when using Google," you know, trusting all those [results] to be reliable…now I definitely feel more comfortable…it's just a different culture in technology that says "of course, you Google it". (Student 1, Lesson 6)

Review of the lesson plans confirmed the curriculum unit's emphasis on self-instructional strategies—e.g., how to study online videos, how to copy/paste code from tutorials in order to experiment with the code, or how to seek out additional references if they did not care for a particular example. As mentioned, student comments stressed that the self-instruction experiences offered by the assignments were most valued. This of course aligns with the study's findings that experience-building must be expanded and prioritized throughout the lesson plans in order to better meet the needs of novices—see Experience before knowledge. (Note: all participants listed interaction with Web-based learning materials among their prior experiences—see Table 2. This means that Web-based materials did not represent the level of novelty that other Web design topics did. Students still reported minimal prior experience and comfort levels, though.)

Implications for Instructors' Decision-Making

Many of the study's conclusions also have implications for instructors' decision-making during implementation, especially for Web design instruction that incorporates constructivist strategies (e.g., student-controlled learning, individualization). In order to address students' myriad of questions on Web design topics, it is of course essential for instructors to be expert in the topics (Bransford et al., 2000), but it is also essential that they be able to effectively model their expertise and conceptual understanding (Royal, 2005). And, to effectively implement constructivist strategies, they also "must act as mentors, facilitators, guides, coaches, and mediators of learning…they must learn how to understand students so that they can interpret responses better, guide communication more effectively, and adjust the help to each student" (Hadjerrouit, 2005, p. 137). For example, many student questions during the study prompted the instructor to use additional or individualized examples and then demonstration of coding those examples—deviations from the lesson plans, in effect. Instructors must be prepared then to respond to student questions by demonstrating typing code related to either the minute details in students' assignment Web pages or to more professional-looking examples that students want to emulate. They must also respond to student questions by demonstrating their reasoning during coding—why use the CSS float property when students are already using CSS alignment properties, for example. The instructor's familiarity with students' assignment code should be established during the assessment process, when the instructor provides feedback on their homework Web pages. Otherwise, it is important for the instructor to have a repertoire of simple, well-designed example Web pages on which to draw, preferably Web pages that novices can emulate to some extent.

Instructors' decisions on which examples to demonstrate or discuss should also be driven by the misunderstandings/errors present in the students' assignment Web pages: novices often are unable to recognize their HTML/CSS misunderstandings (Stepp et al., 2009) and using their code to demonstrate a worked out example serves the dual purpose of prompting students to rethink what is often a shared misunderstanding and concretely showing them how to resolve the error (Schnotz & Kurschner, 2007; Wiggins & McTighe, 2005). Note that students' HTML/CSS errors should not be used as negative examples in front of the group without the student's prior permission. The study's curriculum unit asked students to volunteer their Web pages for discussion and review the week before, for example, and all students must volunteer over the course of the unit, so that students who make more errors are not singled out. As their coding anxiety receded, most study participants asked in-class questions about their errors even when they were not that week's volunteers.

Because the instructor can only address and scaffold for a limited number of errors, they must also continually decide whether to demonstrate or tell an answer or whether to direct learners to resolve errors without assistance. During review and practice sections, the instructor frequently re-posed a student's question to the group and then instructed the group to conduct a Web search to find a resolution (e.g., a tutorial explaining how to achieve a certain CSS style). This allowed the novices to practice self-instructional information-seeking and allowed the group to participate in collaborative problem-solving, which benefits both the help-seeker and their helpers (Park & Wiedenbeck, 2011). Instructors must then be prepared to respond to student questions with conspicuous self-instructional strategies for problem-solving.

The use of HTML/CSS code validators and linters, software that identify typos or code that does not follow syntax or grammar rules, can also be useful for novice coders trying to resolve errors without assistance—again, novices have particular difficulty recognizing their coding mistakes (Stepp et al., 2009). Park et al. (2013) found that if a simple HTML/CSS typo or syntax error was simply pointed out to students, it was enough information for them to fix the error. Validators can then serve that purpose (Hofstetter, 2006), which allows the instructor to focus on more conceptual misunderstandings. Validators may pose difficulty for non-technical learners or novices though, because just like Web design textbooks, validators were designed for experts: "students receive cryptic error messages often referencing lines of code that don't themselves contain the problem, and students must have a good knowledge of the formal structure of the language to decipher these messages and make corrections" (Rosmaita, 2006, p. 270). Ultimately, the instructor must decide whether validators and like tools are appropriate for students or if instructor/peer scaffolding should be relied upon solely. For example, two study participants stopped using Dreamweaver's validator after receiving mostly 'cryptic' warnings messages; they were far more comfortable asking for help in class, where they could count on the instructor's or a peer's simple explanation.

Web design instructors must be able to offer novices simplified explanations of most HTML/CSS examples and scenarios (especially if the instructional materials were written with an Expert Blind Spot), and they may need to explain how a student's far-reaching question connects to their assignment Web pages or at least to the basic topics being introduced. For example, because they interacted with the Internet daily, participants asked multiple questions on Web design topics outside the scope of an introductory course—how to incorporate advanced Youtube.com functionality and the *Wall Street Journal* Web site's complex layout, for example. Establishing student's prior experience and knowledge so that the instructor can determine how simplified their response must be is paramount to instructor's decision-making during implementation, of course (Bransford et al., 2000; Wiggins & McTighe, 2005). This study in particular revealed how failing to establish students' lack of computer literacy knowledge before instruction on Web design begins can be detrimental to learners' understanding (see Computer literacy before Internet literacy section). Pre-assessment of whether students possess an effective model of computers is then crucial (Ben-Ari, 1998; Park & Wiedenbeck, 2011). Discussion or a short question-and-answer section on computer literacy topics during class is likely sufficient for pre-assessing students' understanding of computers. Asking students "what is a browser?" and "how do URLS work?" can reveal if students have ever conceptualized the difference between a browser and an operating system or how file pathing functions, while still seeming relevant to Web design. This implementation's Lesson 1 activity on opening HTML files simultaneously in an Internet browser and in an HTML editor also quickly elicited student questions on files types and software—further indication that the students lacked a conceptual understanding of computers. It is unlikely that non-technical learners such as these participants could self-assess whether they hold an effective model of a computer, and so self-reporting their computer literacy (e.g., on an external survey) may not uncover their misunderstandings. After determining students' levels of computer literacy, the instructor can then make decisions about the extent of computer literacy topics to cover before beginning the in-class file management experience-building activities needed by novices (see Computer literacy before Internet literacy).

Study Limitations

Though this study was an extension of an iterative piloting and curriculum development process, the study's results reported here are ultimately limited to one implementation of the curriculum unit and to one group of participants. A lack of generalizability of the results is then a study limitation. The sample size for this one implementation was also small, which serves as a further threat to the study's external validity: "Threats to external validity are problems that threaten our ability to draw correct inferences from the sample data to other persons, settings, and past and future situations" (Creswell, 2005, p. 293). However, the study's aim to evaluate the curriculum unit as a model for Web design education (i.e., evaluate it based on an authentic implementation), while building on prior data through the use of deductive qualitative analysis, was achieved in detail, regardless of the small number of participants. (Five students still generated/submitted 49 documents for review and analysis, for example.)

The study site also served as a limitation: the curriculum unit was developed and piloted for a 2.5 hour duration, but during the study, the implementation start time was typically delayed. The duration of Lessons 1, 3, 5 and 6 was each close to 2 hours in length, but Lessons 2 and 4 were delayed by close to one hour, with durations of 1:37 and 1:28 respectively. Having one-half to one hour less class-time in which to implement the lesson plans directly impacted the frequency of class ended deviations observed

(ENDC=15.9%—see Figure 5). The key finding that the curriculum unit lacked sufficient in-class experience-building was also impacted by the class-time limitation, e.g., had the lesson plans been afforded additional time, the practice sections located at the end may not have suffered from truncation or omission during the study. The practice time may then have impacted students' experience-building, which in turn may have impacted their misunderstandings. (Previous pilotings that ran the full 2.5 hours did not evidence this, however. Misunderstandings persisted regardless of added time.) The finding that the curriculum unit prioritized knowledge over experience should not be discounted though, since this finding exposed how frequently the curriculum unit did not meet the needs of novices. And, the lack of sufficient class-time emphasized the implementation's authenticity, if nothing else. The lack of finalized Web page content in this particular implementation also likely impacted the results: had participants chosen different Web site projects, student interaction with the Lesson 5 template assignment may have been altered, for example.

The curriculum unit's known limitations should be considered for their impact on the study's conclusions as well. For example, the lesson plans prioritized instruction on HTML and CSS coding over instruction on design (and omitted detailed instruction on Web editing software and computer literacy—see Project Background in chapter 1). And so, a higher frequency of combined student misunderstandings on coding mechanics (e.g., HTML syntax CSS syntax—see Figure 9) than on design concepts (e.g., CSS display, HTML display) simply reflects the emphasis on coding. The study's evidence that the curriculum unit does not provide effective instruction for the topics that it prioritizes still points to a need for reformulation, regardless of a potential limitation.

The potential threats to internal validity associated with the study's qualitative methods and analysis should also be noted: "Internal validity is concerned with how trustworthy the conclusions are that are drawn from the data and the match of these conclusions with reality" (Anfara, et al., 2002, p. 33). However, the use of a mixed-method research design, the triangulation of multiple data collection methods including video recordings, utilizing independent classroom observers, plus a participant–observer, as well as reporting results both quantitatively and qualitatively—all were attempts to strengthen the study's conclusions (Creswell, 2005).

Data collector bias and researcher-as-instrument bias remained a study limitation though: "the qualitative researcher's challenge is to demonstrate that [their] personal interest will not bias the study" (Marshall & Rossman, 1999, p. 28). Again, the mixed-method research design attempted to alleviate potential for generating "personal theory" unsupported by evidence (Marshall & Rossman, 1999, p. 29), as did the use of deductive qualitative analysis, in which the structure of analysis was operationalized on the basis of previous knowledge (Kyngas & Vanhanen, 1999). Evidence of the researcher's Expert Blind Spot (Wiggins & McTighe, 2005) was observed minimally during the study in their role as instructor only, in the form of techno-centric paraphrasing, for example. The study's utilization of multiple levels of analysis, video recordings for data accuracy and non-participants observers for confirming evidence again attempted to alleviate opportunity for this bias (Creswell, 2005), even though the researcher's subjectivity and prior experience studying the curriculum unit, as its designer and developer, must still be recognized as potentially detrimental to objectivity in the study's data collection and analysis.

As discussed in chapter 3, the presence of the researcher, observers and video camera served as reminders that participants were being observed and potential for altered participant behavior during the study—the Heisenberg Effect (Brogan & Biklen, 2003)—should be noted as an additional study limitation.

Future Research

During the course of this study, multiple directions for future research were identified. Not only does the curriculum unit warrant extensive revision to both content and instructional strategies, but it must then be rigorously retested in a manner that is "use-driven strategic research and development focused on issues of improving classroom learning and teaching" (Bransford et al., 2000, p. 250). Supplementary research—external to the curriculum unit—that concretely documents effective instructional strategies for connecting Web design concepts to novices' prior experience is also needed in order to advance Web design education. Again, the new theory of learning argues that "people construct new knowledge and understandings based on what they already know and believe" (Bransford et al., 2000, p. 10)—how then does Web design instruction effectively connect to non-technical learners' lack of prior experiences with the Internet or computer technology? What knowledge or experience should it build on if students have no prior experience? Questions like these have been explored by modern learning theory research for fields like science or reading (Bransford et al., 2000), but there is both a dearth of evidence and little consensus in Web design scholarship (e.g., Dick, 2006; Karper, 2004). Karper's (2004) finding that analysis of other Web sites does not lead students to create effective Web pages was mirrored in part by this study's conclusions that introducing Web design conceptually to novices by connecting it to their prior Internet experiences did not improve student understanding. (For accuracy, this conclusion may only be identified as a failing of this particular curriculum unit implementation.) This conclusion in turn was a replication of Turnley's (2005) finding that exploring Web pages that they cannot imitate ultimately does not help novices connect Web design knowledge to their prior experience.

Research evidencing how to effectively transition non-technical learners and novices from procedural learning to conceptual learning would also address a dearth in Web design scholarship. This curriculum unit strove to employ the instructional strategies recommended by Kalman and Ellis (2007) for teaching Web design conceptually—individualization for students, meaningful assignments, student-controlled learning, self-instructional strategies, instructor as facilitator—but the study results' quantity of deviations and persistent student misunderstandings revealed the complexity involved in employing these strategies. Further research then is warranted to identify better application of these instructional methods.

Summary

Teaching Web design conceptually adds a level of difficulty for students who may already struggle to learn Web design procedurally (Dick, 2006). Web design is an abstract and still-evolving subject, and there is little consensus on how to effectively teach it—conclusions and approaches vary by field. Instruction on CSS adds additional complexity to the scenario by asking students to learn how to achieve visual design using computer languages (Gordon, 2005). In order to evaluate a curriculum unit that strived to apply modern learning theory strategies to the Web design instruction scenario, this study pursued these research questions:

1. What deviations by the instructor occur during implementation of the curriculum unit and why?
2. What student 'misunderstandings' about XHTML and CSS persist throughout curriculum unit implementation and why?

To investigate these questions, a mixed-method research design utilizing classroom observations, video observations, student documents, and the curriculum unit itself (the lesson plans, the website with assignment instructions, etc.) was enacted. Deductive qualitative analysis, in which the student misunderstanding categories collected during pilotings were used as a starting point (see Table 1 in chapter 1), was conducted on the data corpus. A Design-based Research framework also governed the study—in DBR, "researchers assume the functions of both designers and researchers, drawing on procedures and methods from both fields" (Wang & Hannafin, 2005, p. 6). These six key findings about the curriculum unit were identified during analysis:

1. Insufficient experience-building activities during class-time.
2. Insufficient procedural introduction to abstract HTML concepts, like adding tags semantically.
3. Insufficient demonstration of how code translates to display, especially CSS styles.
4. Insufficient procedural instructions for abstract CSS layout concepts, like positioning and flow.
5. Insufficient establishment of the project Web site content and information architecture.

6. Insufficient introduction to computer literacy concepts, especially file pathing.

This study provided evidence that extensive reformulation of the curriculum unit was needed, so that it better meets the needs of non-technical learners and Web design novices—it must be revised so that it prioritizes experience-building over knowledge, which is seen as crucial by the new science of learning (Bransford et al., 2000). The curriculum unit also warrants revision so that it more effectively introduces and builds Web design novices' conceptual learning capabilities. Reformulation and rigorous retesting of the curriculum unit is a first step for further research, as is pursuit of research that better documents how best to instruct for non-technical learners' absence of prior experience. This study's limitation of a lack of generalizability would be addressed by expanded inquiry that addresses the current dearth of research-based Web design education scholarship (Karper, 2004; Kotamraju, 1999).

Web design will continue to be a valued professional and educational activity—"The explosive growth of the Internet has made the knowledge and skills for creating Web pages into general subjects that all students should learn" (Ariga & Watanabe, 2008, p. 815)—and Web design pedagogy must evolve to better employ the standards of modern learning theory so that "focus on the necessary critical and intellectual approaches [is no longer] missing from many Web design curricula" (Royal, 2005, p. 401).

References

Ainsworth, S. (1999). The functions of multiple representations. *Computers & Education*, *33*(2-3), 131–152. doi: 10.1016/S0360-1315(99)00029-9

Ainsworth, S. (2006). DeFT: A conceptual framework for considering learning with multiple representations. *Learning and Instruction*, *16*(3), 183-198. doi: 10.1016/j.learninstruc. 2006.03.001

Andrew, R., & Yank, K. (2008). *Everything you know about CSS is wrong!* Collingwood, VIC, Australia: SitePoint.

Anfara, V. A., Brown, K. M., & Mangione, T. L. (2002). Qualitative analysis on stage: Making the research process more public. *Educational Researcher*, *31*(7), 28-38.

Ariga, T., & Watanabe, T. (2008). Teaching materials to enhance the visual expression of web pages for students not in art or design majors. *Computers & Education, 51*(2), 815-828.

At a glance (n.d.). In *Curry School of Education*. Retrieved April 5, 2012 from http://curry. virginia.edu/about/at-a-glance/

Ben-Ari, M. (1998). Constructivism in computer science education. In D. Joyce & J. Impagliazzo (Eds.), *Proceedings of the 29th SIGCSE Technical Symposium on Computer Science Education (SIGCSE '98)* (pp. 257-261). New York, NY: ACM. doi: 10.1145/273133.274308

Beriswill, J. (2005). Streamlined web design: A study of expert visual designers developing web site interfaces. In P. Kommers & G. Richards (Eds.), *Proceedings* *of World Conference on Educational Multimedia, Hypermedia and Telecommunications 2005* (pp. 916-922). Chesapeake, VA: AACE. Retrieved from http://www.editlib.org/p/ 20196

Berners-Lee, T., & and Connolly, D. (1993, June). *Hypertext Markup Language (HTML): A representation of textual information and metainformation for retrieval and interchange*. Retrieved from http://www.w3.org/MarkUp/draft-ietf-iiir-html-01.txt

Berners-Lee, T., & Fischetti, M. (2000). *Weaving the Web: The original design and ultimate destiny of the World Wide Web*. San Francisco, CA: HarperCollins.

Blackwell, A. F. (2002). First steps in programming: A rationale for attention investment models. In *Human Centric Computing Languages and Environments, 2002. Proceedings. IEEE 2002 Symposia on* (pp. 2-10). IEEE.

Bluttman, K., & Cottrell, L. (2012). *HTML5 Multimedia developer's guide*. New York: McGraw Hill.

Bogdan, R. C., & Biklen, S. K. (2003). *Qualitative research for education: An introduction to theories and methods* (4th ed.). New York: Pearson Education.

Bos, B. (2004). CSS tutorial: Starting with HTML + CSS [Webpage article]. Retrieved from http://www.w3.org/Style/Examples/011/firstcss.en.html

Bransford, J. D., Brown, A. L., & Cocking, R. R. (Eds.). (2000). *How people learn: Brain, mind, experience, and school* (Expanded ed.). Washington, DC: National Academy Press. Retrieved from http://www.nap.edu/openbook.php?isbn=0309070368

Braun, L. W. (2000). Learning from the inside [teaching students how to create their own Web pages]. *Library Journal*, NetConnect Supplement (Summer), 28-30.

Budd, A., Moll, C., & Collison, S. (2009). *CSS mastery: Advanced web standards solutions* (2nd ed.). Berkeley, CA: Apress.

Burch, R. O. (2001). Effective web design and core communication issues: The missing components in web-based distance education. *Journal of Educational Multimedia and Hypermedia, 10*(4), 357-367.

Bureau of Labor Statistics, U.S. Department of Labor. (2013). Graphic designers. *Occupational Outlook Handbook, 2013-14 Edition* [online]. Retrieved from http://www.bls.gov/ooh/

Carter, L. (2006). Why students with an apparent aptitude for computer science don't choose to major in computer science. In *Proceedings of the 37th SIGCSE Technical Symposium on Computer Science Education (SIGCSE '06)* (pp. 27-31). New York, NY: ACM. doi: 10.1145/1121341.1121352

Chafy, R. (1997). Exploring the intellectual foundation of technology education: From Condorcet to Dewey. *Journal of Technology Education, 9*(1), 6-19.

Chou, H., & Wang, Y. (1999). Effects of learning style and training method on computer attitude and performance in world wide web page design training. *Journal of Educational Computing Research, 21*(3), 325-344.

Clark, B. I., Knupfer, N. N., Mahoney, J. E., Kramer, K. M., Ghazali, H., & Al-Ani, N. (1997). Creating web pages: Is anyone considering visual literacy? In *VisionQuest: Journeys toward Visual Literacy. Selected Readings from the Annual Conference of the International Visual Literacy Association* 28th ed. (pp.

354-362). Cheyenne, Wyoming: IVLA. Retrieved from http://eric.ed.gov/?id=ED408990

Coffey, A., & Atkinson, P. (1996). *Making sense of qualitative data: Complementary research strategies*. Thousand Oaks, CA: Sage.

Cohen, L., Manion, L., Morrison, K., & Morrison, K. R. (2007). *Research methods in education* (6th ed.). New York: Routledge.

Collins, A. (1992). Toward a design science of education. In E. Scanlon & T. O'Shea (Eds.), *New directions in educational technology*. Berlin: Springer.

Collison, S. (2006). *Beginning CSS web development: From novice to professional*. Berkeley, CA: Apress.

Connolly, R. (2012). Criticizing and modernizing computing curriculum: The case of the web and the social issues courses. In *Proceedings of the 17th Western Canadian Conference on Computing Education (WCCCE '12)* (pp. 52-56). New York, NY: ACM. doi: 10.1145/2247569.2247587

Creswell, J. W. (2005). *Educational research: Planning, conducting and evaluating quantitative and qualitative research* (2nd ed.). Upper Saddle River, NJ: Pearson Prentice Hall.

Deek, F. P., Coppola, N., Elliot, N., & O'Daniel, N. E. (2000). Cognitive characteristics of web developers: Creativity, meaning construction, & problem solving. *WebNet Journal: Internet Technologies, Applications & Issues, 2*(2), 36-50. Charlottesville, VA: AACE. Retrieved from http://www.editlib.org/p/7996

Descy, D. E. (1999). HTML editors: Web pages in minutes. *TechTrends, 43*(4), 5-7.

Dick, R. F. (2006). Does interface matter? A study of Web authoring and editing by inexperienced Web writers. *Business Communication Quarterly*, *69*(2), 205-215. doi: 10.1177/1080569906069000214

Doering, A., & Veletsianos, G. (2007). Multi-scaffolding environment: An analysis of scaffolding and its impact on cognitive load and problem-solving ability. *Journal of Educational Computing Research*, *37*(2), 107-129.

Döring, N. (2002). Personal home pages on the web: A review of research. *Journal of Computer-Mediated Communication*, *7*(3), *n. pag*. doi: 10.1111/j.1083-6101.2002.tb00152.x Retrieved from http://onlinelibrary.wiley.com/doi/10.1111/j.1083-6101.2002.tb00152.x/full

Ebersole, S. (1997). Cognitive issues in the design and development of interactive Hypermedia: Implications for authoring WWW sites. *Interpersonal Computing and Technology*, *5*(1-2), 19-36.

Feldon, D. F. (2007). Cognitive load and classroom teaching: The double-edged sword of automaticity. *Educational Psychologist, 42*(3), 123-137. doi: 10.1080/00461520701416173

Fiore, M. (1997). Colleges ponder the pros and cons of having students design web sites. *Chronicle of Higher Education*, *43*(47), A221. Retrieved from http://chronicle.com/article/Colleges-Ponder-ProsCons/76711/

Fisher, M. (2001). Design guidelines for optimum teaching and learning on the Web. *Journal of Educational Technology Systems*, *29*(2), 107-118.

Fleiss, J. L., Levin, B., & Paik, M. C. (2013). *Statistical methods for rates and proportions* (3rd ed.). John Wiley & Sons.

Foley, A., & Regan, B. (2002). Web design for accessibility: Policies and practice. *AACE Journal*, *10*(1), 62-80. Retrieved from http://www.editlib.org/p/17779

Garner, S. (2002). Reducing the cognitive load on novice programmers. In P. Barker & S. Rebelsky (Eds.), *Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications 2002* (pp. 578-583). Chesapeake, VA: AACE. Retrieved from http://www.editlib.org/p/10329

Gilbertson, S. (2011, October 26). Discover what's new in CSS 4 [Blog post]. Retrieved from http://www.webmonkey.com/2011/10/discover-whats-new-in-css-4/.

Gilgun, J. F. (2011). Coding for deductive qualitative analysis. *Current Issues in Qualitative Research*, *2*(1), 1-7. Retrieved from http://www.scribd.com/doc/47331325/Coding-in-Deductive-Qualitative-Analysis

Gilgun, J. F. (2005). Deductive qualitative analysis and family theory building. In V. L. Bengston, A. C. Acock, K. R. Allen, P. Dillworth-Anderson & D. M. Klein (Eds.), *Sourcebook of family theory and research* (pp. 83-90). Thousand Oaks, CA: Sage.

Gillenwater, Z. M. (2011). *Stunning CSS3: A project-based guide to the latest in CSS*. Berkeley, CA: New Riders.

Gordon, J. L. (2005). Teaching Hypertext Composition. *Technical Communication Quarterly*, *14*(1), 49-72. doi:10.1207/s15427625tcq1401_5

Hadjerrouit, S. (2005). Designing a pedagogical model for web engineering education: An evolutionary perspective. *Journal of Information Technology Education*, *4*, 115-140.

Hannafin, M. J., & Kim, M. C. (2003). In search of a future: A critical analysis of research on web-based teaching and learning. *Instructional Science*, *31*(4), 347-351. doi:10.1023/A:1024646328974

Hazzan, O. (2004). Mental constructions and constructions of web sites: Learner and teacher points of view. *British Journal of Educational Technology, 35*(3), 323-344.

Hofstetter, F. T. (2006). *Internet literacy* (4th ed.). New York: McGraw-Hill/Irwin.

Howells-Mead, M. (2009, March 3). Ten years of push-button publishing [Blog post]. Retrieved from: http://burson-marsteller.eu/2009/09/ten-years-of-push-button-publishing/

HTML. Retrieved July 9, 2013 from http://en.wikipedia.org/wiki/HTML

Jakovljevic, M., Ankiewicz, P., de Swardt, E., & Gross, E. (2004). A synergy between the technological process and a methodology for web design: Implications for technological problem solving and design. *International Journal of Technology & Design Education*, 14(3), 261-290. doi: 10.1007/s10798-004-4868-5

Kahn, S. (2011, March). Salman Khan: Let's use video to reinvent education [Video file]. Retrieved from http://www.ted.com/talks/lang/eng/salman_khan_let_s_use_video _to_ reinvent _education.html

Kalman, H. K., & Ellis, M. L. (2007). Improving undergraduate computer instruction: Experiments and strategies. *AACE Journal*, *15*(1), 23-38.

Karper, E. E. (2004). *"Ordinary people do this": A rhetorical case study of novice web design.* (Doctoral thesis). (UMI No. 3150785).

Keller, J. (2010). *Motivational design for learning and performance: The ARCS model approach*. New York: Springer.

Koehler, M. J., & Mishra, P. (2005). What happens when teachers design educational technology? The development of technological pedagogical content knowledge. *Journal of Educational Computing Research*, *32*(2), 131-152.

Kotamraju, N. P. (1999). The birth of web site design skills: Making the present history. *American Behavioral Scientist, 43*(3), 464-474. doi:10.1177/00027642990430003009

Krunić, T., Ružić-Dimitrijević, L., Petrović, B., & Farkaš, R. (2006). Web design curriculum and syllabus based on Web design practice and students' prior knowledge. *Journal of Information Technology Education*, *5*, 317-335. Retrieved from http://eric.ed.gov/? id=EJ808955

Kyngas H., & Vanhanen L. (1999). Content analysis. *Hoitotiede [Nursing Science], 11*(1), 3–12

Landis, J. R., & Koch, G. G. (1977). The measurement of observer agreement for categorical data. *Biometrics*, *33*(1), 159-174.

Lawless, K. A., & Brown, S. W. (2003). Introduction: From digital dirt road to educational expressway: Innovations in web-based pedagogy. *Instructional Science, 31*(4), 227-351. doi: 10.1023/A:1024665423522

Leahy, M., & Twomey, D. (2005). Using web design with pre-service teachers as a means of creating a collaborative learning environment. *Educational Media International*, *42*(2), 143-151.

Leahy, W. M., Chandler, P., & Sweller, J. (2003). When auditory presentations should and should not be a component of multimedia instruction. *Applied Cognitive Psychology*, *17*(4), 401-418. doi: 10.1002/acp.877

Lenhart, A., Purcell, K., Smith, A., & Zickuhr, K. (2010). *Social media and mobile Internet use among teens and young adults.* Pew Internet and American Life Project report. Retrieved from http://www.pewinternet.org/Reports/2010/Social-Media-and-Young-Adults.aspx

Lever-Duffy, J., McDonald, J. B., & Mizell, A .P. (2005). *Teaching and learning with technology* (2nd ed.). Boston: Pearson Education.

Lie, H. W., & Bos, B. (1999). *Cascading Style Sheets: Designing for the Web* (2nd ed.). Harlow, England: Addison Wesley Longman.

Lie, H. W., & Saarela, J. (1999). Multipurpose Web publishing using HTML, XML, and CSS. *Communications of the ACM*, *42*(10), 95-101.

Lim, B. R., Plucker, J. A., & Bichelmeyer, B. (2003). Learning by web design: How it affects graduate student attitudes. *College Teaching*, *51*(1), 13-19.

Liu, C., & Downing, C. (2010). Using Cascading Style Sheets to Design a Fly-Out Menu with Microsoft Visual Studio. *Journal of Information Systems Education*, *21*(3), 275-81.

Mackey, T. (2005). Producing Web Based Multimedia for Web Design Instruction. In P. Kommers & G. Richards (Eds.), *Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications 2005* (pp. 3240-3246). Chesapeake, VA: AACE. Retrieved from http://www.editlib.org/p/20581

Mackey, T. P., & Ho, J. (2005). Implementing a convergent model for information literacy: Combining research and web literacy. *Journal of Information Science, 31*(6), 541-555. doi:10.1177/0165551505057018

Maddux, C. D., Liu, L., Cummings, R., & Smaby, M. (2008). A totally online course in web design for teachers: Aids and cautions. In K. McFerrin et al. (Eds.), *Proceedings of Society for Information Technology & Teacher Education International Conference 2008* (pp. 4227-4232). Chesapeake, VA: AACE. Retrieved from http://www.editlib.org/p/27919

Marshall, C., & Rossman, G. B. (1999). *Designing qualitative research* (3rd ed.). Thousand Oaks, CA: Sage.

Marx, S. (2003). Teaching hypermedia authoring to teachers: An evolutionary curriculum. In A. Rossett (Ed.), *Proceedings of World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education 2003* (pp. 1100-1103). Chesapeake, VA: AACE. Retrieved from http://www.editlib.org/p/12543

Miller, C. S., Perković, L., & Settle, A. (2010). File references, trees, and computational thinking. In *Proceedings of the 15th Annual Conference on Innovation and Technology in Computer Science Education (ITiCSE '10)* (pp. 132-136). New York, NY: ACM. doi: 10.1145/1822090.1822128

Mills, C. (2013). *Practical CSS3: Develop and design*. Berkeley, CA: PeachPit.

Mull, R. F. (2001). *Teaching web design at the higher education level.* (Master’s thesis). Retrieved from ProQuest Dissertation and Theses database. (UMI No. 1404655).

Newby, T. J., Stepich, D. A., Lehman, J. D., & Russell, J. D. (2006). Theory into application. *Educational technology for teaching and learning* (3rd ed.). Upper Saddle River, NJ: Merrill.

Nielsen. J. (2003). Usability 101: Introduction to usability [Webpage article]. Retrieved from http://www.useit.com/alertbox/20030825.html

Notess, G. R. (2006). Diverging web markup choices. *Online (Weston, Conn.)*, *30*(6), 43-45.

Niess, M. L., Lee, J. K., & Kajder, S. B. (2008). *Guiding learning WITH technology*. Hoboken, NJ: Wiley & Sons.

Park, T. H., Saxena, A., Jagannath, S., Wiedenbeck, S., & Forte, A. (2013). Towards a taxonomy of errors in HTML and CSS. In *Proceedings of the 9th Annual International ACM Conference on International Computing Education Research (ICER '13)* (pp. 75-82). New York, NY: ACM. doi: 10.1145/2493394.2493405

Park, T. H., & Wiedenbeck, S. (2010). First steps in coding by informal web developers. In *Proceedings of the 2010 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)* (pp. 79-82). IEEE. doi: 10.1109/VLHCC.2010.20

Park, T. H., & Wiedenbeck, S. (2011). Learning web development: Challenges at an earlier stage of computing education. In *Proceedings of the 7th International Workshop on Computing Education Research (ICER '11)* (pp. 125-132). New York, NY: ACM. doi: 10.1145/2016911.2016937

Parkinson, A., & Redmond, J. A. (2002). Do cognitive styles affect learning performance in different computer media? In *Proceedings of the 7th Annual Conference on*

Innovation and Technology in Computer Science Education (ITiCSE '02) (pp. 39–43). Aarhus, Denmark: ACM. doi: 10.1145/544414.544427

Pope C., Ziebland S., & Mays N. (2000). Qualitative research in health care: analysing qualitative data. *British Medical Journal*, *320*, 114-116.

Powell, T. A. (2002). *Web design: The complete reference* (3rd ed.). Berkeley, CA: Osborne McGraw-Hill.

Powell, T. A. (2010). *HTML & CSS: The complete reference* (5th ed.). New York: McGraw-Hill.

Raggett, D., Lam, J., Alexander, I., & Kmiec, M. (1998). *Raggett on HTML 4.* Harlow, England: Addison Wesley Longman.

Raths, J. (2002). Improving instruction. *Theory into Practice*, *41*(4), 233-237.

Reber, R. (2005). Assessing motivational factors in educational technology: The case of building a web site as course assignment. *British Journal of Educational Technology, 36*(1), 93-95. doi:10.1111/j.1467-8535.2004.00441.x

Reimann, P. (2011). Design-based research. In L. Markauskaite, P. Freebody & J. Irwin (Eds.), *Methodological choice and design: Scholarship, policy and practice in social and educational research* (pp. 37-50). Springer Netherlands. doi: 10.1007/978-90-481-8933-5_3

Reimann, P. (2010, November). *Design-based research in the learning sciences* [PowerPoint slides]. Retrieved from http://www.slideshare.net/perei/designbased-research-an-introduction

Robson, E., & Freeman, E. (2012). *Head first HTML and CSS* (2nd ed.). O'Reilly Media.

Rockwell, S. (2008). Working smarter, not harder: Reaching the tough to teach. *Kappa Delta Pi Record*, *44*(3), 108-113.

Rosmaita, B. J. (2006). Accessibility first! A new approach to web design. *ACM SIGCSE Bulletin*, *38*(1), 270-274.

Royal, C. (2005). Teaching web design in journalism and mass communications programs: Integration, judgment, and perspective. *Journalism and Mass Communication Educator*, *59*(4), 400-414.

Schnotz, W., & Kürschner, C. (2007). A reconsideration of cognitive load theory. *Educational Psychology Review*, *19*, 469–508. doi: 10.1007/s10648-007-9053-4

Selber, S. A. (2004). Reimagining the functional side of computer literacy. *College Composition and Communication*, *55*(3), 470–503.

Shank, G. D. (2002). *Qualitative research: A personal skills approach.* Upper Saddle River, NJ: Merrill Prentice Hall.

Sklar, J. (2008). *Principles of web design* (4th ed.). Cambridge, MA: Course Technology, Inc.

Smaldino, S. E., Russell, J. D., Heinich, R., & Molenda, M. (2005). *Instructional technology and media for learning* (8th ed.). Upper Saddle River, New Jersey: Pearson Prentice Hall.

Stepp, M., Miller, J., & Kirst, V. (2009). A “CS 1.5” introduction to web programming. In *Proceedings of the 40th ACM Technical Symposium on Computer Science Education (SIGCSE ‘09)* (pp. 121-125). New York, NY: ACM. doi: 10.1145/1508865.1508908

Storey, D. (2013, July). Why should I learn HTML or CSS? Everything I need is in DreamWeaver or a book [Blog post]. Retrieved from http://demosthenes.info/blog/ 296/Why-Should-I-Learn-HTML-or-CSS-Everything-I-Need-Is-In-DreamWeaver-Or-A-Book

Storey, D. (2013, September). Starting Your Web Development Education Online [Blog post]. Retrieved from http://demosthenes.info/blog/571/Starting-Your-Web-Development-Education

Taylor, M. J., Salces, F. S., & Duffy, S. P. (2005). Teaching web site design: Science or art? *Journal of Further & Higher Education, 29*(4), 331-340. doi: 10.1080/03098770500353425

Teague, J. C. (2006). *CSS, DHTML, and Ajax: Visual quickstart guide* (4th ed.). Berkeley, CA: Peachpit Press.

Tilley, C. L., & Collison, D. (2007). New mentors for new media: Harnessing the instructional potential of cognitive apprenticeships. *Knowledge Quest: Journal of the American Association of School Librarians*, *35*(5), 26-31.

Tuck, M. (2011, March 28). From nothing to something: Story of a self-taught designer [Webpage article]. Retrieved from http://sixrevisions.com/web_design/from-nothing-to-something-story-of-a-self-taught-designer/

Turnley, M. (2005). Contextualized design: Teaching critical approaches to web authoring through redesign projects. *Computers and Composition, 22*(2), 131-148. doi: 10.1016/j.compcom.2005.02.007

Victor, S. (2002). User-centered web site design in the instructional technology curriculum. In D. Willis et al. (Eds.), *Proceedings of Society for Information*

Technology & Teacher Education International Conference 2002 (pp. 1006-1009). Chesapeake, VA: AACE. Retrieved from http://go.editlib.org/p/10657

W3C.org. “HTML & CSS.” Retrieved from http://www.w3.org/standards/webdesign/htmlcss

Walker, K. (2002). Theoretical foundations for website design courses. *Technical Communication Quarterly*, *11*(1), 61-83. Retrieved from http://eric.ed.gov/?id=EJ646978

Wang, F., & Hannafin, M. J. (2005). Design-based research and technology-enhanced learning environments. *Educational Technology Research and Development, 53*(4), 5-23.

Wang, X., & McKim, J. C. (2013). The opportunities and challenges to teach web programming in computer science curriculum CS2013 [Consortium for Computing Sciences in College 2013]. *Journal of Computing Sciences in Colleges*, *29*(2), 67-78.

Web design. (2013). Retrieved July 9, 2013 from http://en.wikipedia.org/wiki/Web_design

Wempen, F., Chase, N., Jacobs, K., McCall, K., Nielsen, J. J., & Schmid, P. (2007). Chapter 31: Developing web content. In *Special Edition Using Microsoft® Office Word 2007* (pp. 857-874). Indianapolis, IN: Que Publishing.

Weyl, E. (2012). *What's new in CSS3* [E-book]. O'Reilly Media. Retrieved from http://proquest.safaribooksonline.com/book/-/9781449356828

Whitehead, J. (2002). A proposed curriculum for a masters in web engineering, *Journal of Web Engineering, 1*(1), 18-22. Retrieved from http://www.soe.ucsc.edu/~ejw/papers/whitehead-webeng.pdf

Wiggins, G., & McTighe, J. (2005). *Understanding by design* (Expanded 2nd ed.). Alexandria, VA: ASCD.

Wilcox, M. (2008, December 9). The fundamental problems with CSS3 [Webpage article]. Retrieved from http://2002-2012.mattwilcox.net/archive/entry/id/1031/

Wilton-Jones, M.T. (2011, March 19). Browser problems [Webpage article]. Retrieved from http://www.howtocreate.co.uk/tutorials/css/problems

Zhang, P., & von Dran, G. M. (2000). Satisfiers and dissatisfiers: A two-factor model for website design and evaluation. *Journal of the American Society for Information Science, 51*(14), 1253-1268. doi:10.1002/1097-4571(2000)9999:9999<::AID-ASI1039>3.0.CO;2-O

Appendix A

Sample External Review Panel Questionnaire

< xhtml & CSS }

Lesson 3 Review

INSTRUCTIONS: Respond to all questions, indicating to what degree the lesson plan meets the criteria. Then briefly detail the reason(s) for your rating, indicating what further information is needed, please.

---

1. How appropriate is the lesson plan for instructors at the post-secondary level?

   [ ] Completely [ ] Substantially [ ] Somewhat [ ] Not at all

   Please describe in what way the lesson plan is not appropriate:

2. How well does the lesson plan provide directions that ensure effective implementation and management?

   [ ] Completely [ ] Substantially [ ] Somewhat [ ] Not at all

   Please identify where in the lesson plan the directions are inadequate.

3. How well does the lesson plan provide directions that ensure clear communication with and complete instructions to students?

   [ ] Completely [ ] Substantially [ ] Somewhat [ ] Not at all

   Please identify where in the lesson plan the directions are inadequate.

4. How clear are the lesson's learning objectives or the purpose of the lesson and assignment?

   [ ] Completely [ ] Substantially [ ] Somewhat [ ] Not at all

   Please identify where in the lesson plan clarity is needed.

5. How well does the lesson align with its learning objectives?

   [ ] Completely [ ] Substantially [ ] Somewhat [ ] Not at all

   Please describe in what way the lesson does not align:

6. How appropriate are the instructional strategies for the lesson's learning objectives?

   [ ] Completely [ ] Substantially [ ] Somewhat [ ] Not at all

   Please describe in what way the strategies are not appropriate:

7. How well does the lesson use strategies that actively address a range of student learning styles?

   [ ] Completely [ ] Substantially [ ] Somewhat [ ] Not at all

   Please describe in what way learning-style differences are not addressed:

8. How well does the lesson use strategies that actively address a range of student interests?

   [ ] Completely [ ] Substantially [ ] Somewhat [ ] Not at all

Please describe in what way the lesson is not student-centered:

9. How well does the lesson require students to process information actively, rather than passively?

[ ] Completely [ ] Substantially [ ] Somewhat [ ] Not at all

Please describe in what way the lesson does not allow for active processing:

10. How well does the lesson promote sustained involvement and attention from post-secondary students?

[ ] Completely [ ] Substantially [ ] Somewhat [ ] Not at all

Please describe in what way the lesson does not promote involvement:

11. How appropriate are the lesson's instructional methods for the cognitive levels of post-secondary students?

[ ] Completely [ ] Substantially [ ] Somewhat [ ] Not at all

Please describe in what way the methods are not appropriate:

12. How appropriate are the lesson's instructional materials for the cognitive levels of post-secondary students?

[ ] Completely [ ] Substantially [ ] Somewhat [ ] Not at all

Please describe in what way the materials are not appropriate:

13. How well does the lesson allow students with a wide range of readiness levels to demonstrate their knowledge or understanding?

[ ] Completely [ ] Substantially [ ] Somewhat [ ] Not at all

Please describe in what way the lesson does not allow demonstration:

14. How well does the lesson allow students with a wide range of readiness levels to demonstrate their skills?

[ ] Completely [ ] Substantially [ ] Somewhat [ ] Not at all

Please describe in what way the lesson does not allow demonstration:

15. How reasonable are the lesson's technology set-up and interaction requirements for instructors?

[ ] Completely [ ] Substantially [ ] Somewhat [ ] Not at all

Please describe in what way the technology requirements are not reasonable:

16. How reasonable are the lesson's technology interaction requirements for post-secondary students?

[ ] Completely [ ] Substantially [ ] Somewhat [ ] Not at all

Please describe in what way the technology requirements are not reasonable:

17. How unbiased is the lesson towards race/ethnicity, gender, socioeconomic status, geographic location and disability?

[ ] Completely [ ] Substantially [ ] Somewhat [ ] Not at all

Please describe in what way the lesson is biased:

18. Any additional feedback or comments for improving or revising the lesson plan?

Appendix B

Sample Observation Instrument

 OBSERVER ID: ______________________

Observation Key & Examples:

CODE KEY: **Cov**: Covered / **Mod**: Modified / **MRepl**: Replaced / **Ret**: Returned to / **Omit**: Omitted / **NotD**: Not Discernible

Covered: Topic and activities were addressed to the extent (and sequence) indicated in the lesson plan. Paraphrasing is likely and OK.

Modified: Topic was addressed but the explanation or activity was expanded or changed. The examples were altered or more were used.

Replaced: Topic or activities were replaced with information or strategies not in the lesson plan. Lecture isn't working, so instructor replaces it with demonstration.

Returned to: Topic was addressed after being skipped, was reviewed twice, or was addressed out of sequence.

Omitted: Topic or activity was left out and not returned to.

Not Discernible: Topic or activity could not be observed. Class ended or observer's view was obstructed.

The study's goal is to evaluate the lesson plans in an authentic implementation. This is done by tracking the instructor's deviations from the lesson plans and identifying the reasons for them. Students' misunderstandings (questions) should also be tracked to aid in identifying the reasons behind the deviations. Take notes when either are observed, please. Examples:

D: Instructor skips list tag section
D: Instructor uses different explanation based on student asking "why?"
D: Instructor skips question prompt during lecture.
D: Student answers email during demonstration; asks Instructor to model code again.

Q: Why doesn't my web page work?
D: Student's webpage used as problem-solving example; Review session returned to
D: Source rendering discussion from prior lesson returned to
Q: Do I have to use columns?
D: CSS layout lecture returned to
D: Review session returned to…more student questions

Lesson 1: Introduction to HTML/XHTML

Preparation:	*Please see the Unit Overview for detailed student or classroom computer requirements.*		
Materials:	1. Unit Outline handout (1 per student) • This info may be posted to a class website, blog or forum post, etc. 2. Example website(s) with clean XHTML code 3. Example free CSS template from http://www.freecsstemplates.org 4. HTMLDog.com HTML Tag Reference: http://www.htmldog.com/reference/htmltags/ 5. XHTML vs HTML tutorial from W3Schools.com: http://www.w3schools.com/html/html_xhtml.asp 6. Assignment instructions webpage (on the class website or a blog entry, forum post, etc.) • Links to the assignment's tutorials, videos, and readings must be clickable 7. Demonstration Code file (demonstrate_L1.html)		
Objectives:	By building the Lesson 1 assignment webpage, students will demonstrate that they can • Type HTML and XHTML tag syntax accurately ○ Follow XHTML syntax guidelines/rules for tag pairing, closing and nesting ○ Follow HTML guidelines/rules for tag placement based on meaning (e.g., a paragraph in a <p>) ○ Identify required structural HTML tags (<html>, <head>, <body>, etc.) ○ Distinguish between functional sections of an HTML document (<head> vs. <body>) ○ Distinguish between font, list and link tags ○ Activate hyperlink functionality (links work in browser) ○ Path to an <img> (image displays in browser) • Distinguish between content (display) and HTML code (structure) ○ Identify the parts of a webpage displayed in a browser (header, navigation, footer, title, etc.) ○ Identify the parts of a webpage not displayed in a browser (code, tags, etc.)		
Sequence:		**Notes:**	
Informal Preassess ment:	*Let's see a show of hands…* 1. *How many of you have built or edited webpages before?* Prompt for details…unless no hands.	Lesson plan piloting revealed that prior knowledge of programming, systems administration or other advanced computer skills *did not* improve HTML/CSS comprehension. These students	Cov Ret x Mod Omit MRepl NotD Notes:

	2. *What kind of online content have you posted, if not HTML? What kind of 'push-button publishing' have you done?* ○ Examples: Blogs, wikis, comments, photos/videos, Facebook 3. *What kind of Web-based materials have you studied before?* ○ Examples: tutorials, videos, websites, distance education… Prompt for details… 4. *You actually have prior experience with Web sites, since you use them daily in a myriad of ways. We will add technical skill and conceptual understanding to your prior experience.* 5. *There exists now an expectation that anyone can post content to the Web. It's a communication skill, not just a technology skill. This unit attempts to improve your understanding of how Web pages work and how they are created!* 6. *Please note that this is not software training. You will learn the concepts behind HTML, coding, and Web design. You will also gain strategies for studying technology. Web design is an enormous topic—too large to memorize and practice in just 6 lessons—so self-instruction skills will be emphasized.*	were comfortable typing code or posting to Web servers, but their assignments' accuracy and decision-making levels were the same as non-technical students. Direct prior knowledge of HTML and CSS *did* improve students' ability to complete the assignments, though some still struggled with XHTML syntax or rules (accuracy).	
Unit Outline & Schedule:	Direct students to the schedule on the Unit Outline handout. . Open an example free CSS template at http://www.freecsstemplates.org Review **the front** of the Unit Outline handout as needed: 1. *Over the next 6 lessons/classes you will learn to build a simple Web site. Expect lots of editing time in this class, because you will revise and repair your Web pages, turning them in multiple times. Expect detailed feedback and comments on your assignments, too.*	Note: Introductions to HTML-editing software like Dreamweaver or NVU should be a separate lesson. (See Unit Overview for more info and links to tutorials.)	Cov — Ret Mod — Omit MRepl — NotD Notes:

	2. *We start from scratch with an introduction to HTML tags and creating a simple Web page. We will then build a full website by adding navigation, layout and content like images or multimedia. We will finish by downloading free graphics-rich templates.* 3. *We will spend the most time on Cascading Style Sheets (CSS) because of its complexity. CSS is the computer language that contains the formatting and styling information for the Web page.* 4. *You will also write 3 brief discussion responses and read or watch tutorials about HTML, CSS and Web pages.* Prompt for questions…	CSS further defined: Cascading Style Sheets are external files that you link to from your HTML. CSS contain all styling and formatting and must not contain structure or content.	
Line-Item Feedback:	5. *Notice on the* Unit Outline handout *that you must REPAIR each webpage assignment after receiving instructor's feedback*. ***Repairs are REQUIRED and part of your final portfolio****! Line-item feedback and repairs will be discussed fully in Lesson 2, after you submit your first webpage.*		
Internet & Browser Literacy:	Open an example Web site like Empty Oceans (http://www.pbs.org/emptyoceans/) or ask students for example sites to discuss. *When I talk about Web design, I use acronyms and terminology–stop me if I use a word you don't know. I will repeatedly refer to the visual or organizational parts of a Web page, as well as Web browser parts.* **1. Parts of an Internet Browser:** *How do Web browsers work? (They're software, after all.) How do Web pages work in browsers?* • Browsers display Web pages and other digital files saved on Web servers. They translate computer languages into visual display and functionality. Discuss the browser frame, window and tab functionality, as needed	Instructors should choose example websites different from those in the lesson plans. Sharing your favorite websites (and your students') invites engagement. At minimum, instructors should choose websites with simple, clean and correct XHTML code.	Cov Ret Mod Omit MRepl NotD Notes:

	2. What is a Web page? *What are the displayed parts of a Web page? Web pages often have visual or organizational sections—what are they?* • Header, title – identifying info • Menu, site-wide navigation, utilities • Content area, body, left-column, right-column • Footer *Remember these parts of a Web page. You will see them repeatedly, and you will create them on your own Web pages. Notice also the Web page's layout: it has rows and columns.* Prompt for questions…	If needed: Mention to students that they should upgrade to the latest versions of FF, IE, etc. Mention that webpages view differently in different browsers. It will be easier if students use the same browsers.	
Intro to HTML and XHTML:	View the example Web page's **page source** (Right-click>View Page Source) in a separate browser tab, or side-by-side. Open http://www.htmldog.com/reference/htmltags/ for reference. *We have just talked about the visual parts of a Web page, so that we can now talk about the code. If we look 'under the hood' we see that the Web page is built using HTML code known as tags.* **1. HTML Defined:** *HTML stands for HyperText Markup Language. HTML is a computer language, but not a programming language. It is code/language that is processed or translated by Web browser software. (The code does not display.)* *HTML 'marks up' your content (text and images) to tell the browser how to display it. HTML tags are like punctuation, but for presentation, not just sentence grammar. You use presentational punctuation every time you write a sentence; now you'll learn how to use HTML to make it.* 2. *What kinds of tags does a Web page have?*	Note: Written examples of presentational punctuation include page numbers or capitalizing the first word of a sentence.	Cov / Ret Mod / Omit MRepl / NotD Notes:

<table>
<tr>
<td></td>
<td>• Structure tags: <html> <head> <body>
• Display tags – anything in the <body>
    ○ Text tags: <h1>, <p>, <ul>,
    ○ Multimedia or content tags: <img>
    ○ Hyperlink tags: <a>, <link>

(Refer to http://www.htmldog.com/reference/htmltags/ if needed.)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>3. HTML/XHTML guidelines and rules:

Using the page source opened earlier, review HTML guidelines. View other example Web sites, if there’s time or student interest…How does HTML work?

a. Every HTML Web page starts/ends with <html></html>
b. Every Web page contains a HEAD and a BODY
    • The <head> holds code that is processed but not displayed
    • The <body> holds displayed content
c. Add tags based on meaning, i.e., semantics
    • A paragraph of text should be placed inside a <p>, not an <h1>
    • A list should be placed in an <li>, not a <p>
d. Tags come in pairs with an open/close or start/end
    • <body></body>
e. Unpaired tags use a slash to close (XHTML guideline)
    •
, <img />, <link />, <hr />
f. Some tags only appear once, others repeat
    • Structure tags appear once, e.g. <html>,<head>, <body>
    • Display tags repeat to mark up content, e.g. <p>
g. Tags appear in specific locations, in specific orders
    • Hidden tags like <title> go in the <head>
    • Display tags go in the <body>
    • List item tags <li> only go inside ordered or unordered list tags: <ol> or
    • Tags must open/close in the correct, mirrored order:</td>
<td>If needed:

Mention that some tags do not work in all browsers. Example: the quote tag <q> doesn’t work in IE.</td>
<td>| | Cov | | Ret |
| | Mod | | Omit |
| | MRepl | | NotD |

Notes:</td>
</tr>
</table>

<table>
<tr>
<td></td>
<td>• No: <h2> subtitle <p>text</h2></p>
• Yes: <h2>subtitle</h2><p>text</p>
h. Most tags should NOT be nested
• Don’t place a PARAGRAPH inside a <li>, <h1> or another <p>
• Avoid placing a <ul>, <ol> or <img /> inside a <p>
• Exception: Place LINKS inside paragraphs or LIST ITEMS inside <ul> or

Note: XHTML stands for Extensible Hypertext Markup Language. It is an updated, “cleaner” version of HTML. I’ll use the two terms synonymously. Your Lesson 1 homework will be to learn about HTML and XHTML. ***Use XHTML syntax and guidelines in your Web pages!***

Prompt for questions…</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Create a Simple Webpage:</td>
<td>**1. Briefly go over the full Lesson 1 assignment instructions and prompt for questions:**

1. Complete the tutorials at W3Schools.com and HTMLDog.com
  a. Welcome to web-based learning!
2. Watch the tutorials on Youtube.com
3. Build a simple webpage using the tags you just learned
  a. Add ‘About Me’ text introducing yourself
  b. Save the HTML document to your web space and email its full-path web address
4. Bring questions for next lesson’s class discussion

Review sessions are based on your questions and issues with the homework or tutorials—so bring them!

Note: When you type your ‘About Me’ text, try to follow the examples in the Improve the Readability of Your Webpage *article.*

2. Walk through the W3schools XHTML vs. HTML tutorial as a class. Direct students to</td>
<td>NOTE: students may need additional guidance on how to study the online content like videos and tutorials. Piloting revealed that some students had no experience with the tutorial process and defaulted to traditional study habits, which does not work in this Web-based learning scenario.

Piloting also revealed that XHTML syntax (
) and guidelines, like nesting, were among the most difficult topics. This is why the XHTML vs HTML tutorial is chosen as the in-class example. The XHTML guidelines will need emphasis and review each week.</td>
<td>Cov Ret
Mod Omit
MRepl NotD

Notes:</td>
</tr>
</table>

	1. Click on W3Schools.com tutorials link: [XHTML vs HTML] 2. Review the XHTML guidelines and code examples: • All tags must be properly nested • All tags must be closed • Unpaired tags must have an end slash, e.g. 3. Explore TryIt Editor format—revise code on left, see display outcome on right 4. Review other W3schools tutorials as needed, or show how they can copy/paste their sample code into an HTML editor 3. *Notice that the Webpage assignment is side-by-side with the tutorials and videos—this is intentional. Find a study pattern for the Web-based materials that is most useful to you. If you want to build the webpage while following along with videos and tutorials, do so! This is not a traditional read-the-chapter, do-the-assignment situation.* Prompt for questions… *As you follow the tutorials, you will be introduced to the XHTML tags needed to build a simple webpage. Use one of the online editors or an HTML editor (Notepad, Dreamweaver or SublimeText) to type your HTML.*	If needed: Locate a text editor on a PC at: Start Menu > All programs > Accessories > Notepad or Wordpad In Notepad, Save As… about.html by simply typing "about.html". Ignore the prompts to save as a .TXT file.	
	1. Demonstrate Typing HTML: Prompt students to copy the Lesson 1 Demonstration Code file and follow along on their computers, if wished. 1. Create a new 'website folder' on the desktop 2. Open an HTML editor (e.g., Kompozer, Notepad, Dreamweaver) 3. Open the Lesson 1 Demonstration Code file (or a new HTML document) 4. Type or retype the demonstration code • Use HTML comments to hide the sample code, if wished 5. Save As… about.html to the website folder 6. View the **local** about.html in a browser 7. Upload the HTML document to a web space	Piloting revealed that students benefited from following along in the Demonstration Code on their laptops, not just the overhead. Most opened their own copy and typed changes, asking questions along the way. If needed: Confirm that students can access their personal web spaces to post/view websites (including images). This will require a separate lesson if they have not	Cov Ret Mod Omit MRepl NotD Notes:

<table>
<tr>
<td></td>
<td>8. View the remote about.html in a browser
9. Test the image at its full-path</td>
<td>used web spaces before. (See Unit Overview for more info.)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2. Pathing/Hyperlinking:

How do links work? How you use links on a webpage? What does a link path (URL) mean?

a. Associative: text or visual link to related info
b. Hierarchical: organizational link denoting information architecture (the menu)

Relative vs. Full-Path:
c. Relative means ‘relative to your website’, in the same folder or location
d. Full-path when the link (or image) is outside your website
e. Link paths mirror the folder structure of your web space

Examples:
f. Full-path: <a href=“http://www.website.com/index.html”>
g. Relative: <a href=“index.html”>
h. Full-path:
i. Relative:
j. Relative, in a higher folder: <img src=“../images/photo.jpg”
k. ../ equals the UP ARROW.

Prompt for questions…</td>
<td></td>
<td>Cov Ret
Mod Omit
MRepl NotD

Notes:</td>
</tr>
<tr>
<td></td>
<td>3. Index.html:

• Index.html, home.html, default.htm are standard index filenames, meaning browsers are programmed to display them as the first webpage in a folder.
• If you don’t specifiy a file name as a URL, the browser shows index.html . Example:
• http://www.virginia.edu vs http://www.virginia.edu/index.html</td>
<td></td>
<td>Cov Ret
Mod Omit
MRepl NotD

Notes:</td>
</tr>
</table>

	Note: HTML files names have requirements: No spaces, no special characters, and no long, convoluted names! *Note also that Microsoft Word's HTML is BAD—don't save from Word! Don't type your code in Word! Use Notepad or Kompozer instead.* **4. HTML Comments:** <!-- Comments are greyed out and do not display in the browser window. This is what your feedback will look like. -->		
	5. Problem-Solving: *Since you are hand-typing your first HTML assignment, be prepared for typos. Accuracy is required when you code webpages (by browsers, not the instructor), but typos are common. Also, make sure that your content (images), displays.* *So,* ***always test/view your webpages*** *in Internet browsers to make sure they work. If they don't work (look/display correctly), revise your HTML.* 1. Do your tags open and close in pairs? 2. Do your unpaired tags have a closing slash (XHTML)? 3. Was your image uploaded to the web space? a. Type in it's full-path in the browser to test 4. Is the path to the image correct? 5. Does it have the right file name: index.html? 6. Is it in the right folder: /public_html/? 7. Did you remove the default homepage: home.html?		Cov Ret Mod Omit MRepl NotD Notes:
	6. Deprecated Tags: *Some HTML tags are 'deprecated'. There are different versions of HTML and XHTML. HTML has been updated and improved (v. 5). This means some tags have been phased out. Example: <center> and <font> were replaced by CSS.* ***Don't use the <center> or <font> tag!***		Cov Ret Mod Omit MRepl NotD Notes:

	Prompt for questions…Discuss code-view versus design-view or source rendering concepts as needed.		
Wrap-Up:	**Ask for Volunteers:** *Next lesson, we will discuss your webpages (including code) during the review session. Volunteers, please?* (At least 3) *If you do not volunteer your webpage this lesson, you must do so for a later lesson.* Remind students that they need: • HTML editor, like Notepad, Wordpad or Text Edit (Mac) • Web space for posting their webpages • Updated Internet browsers • Optional: Dreamweaver software or other recommended editing software (Kompozer). Download the free trial. Email instructor if there are issues with the assignment. ASK FOR HELP! • All questions and feedback are welcome *Bring your questions to class for the review session!*	Piloting revealed student anxiety over the class critiquing their webpages. Asking for volunteers in advance reduced anxiety, by giving them a week to get used to the idea. Some students were also encouraged to seek help in advance, knowing that the class would look at their code.	Cov Ret Mod Omit MRepl NotD Notes: