Keeping Wonder in Check: Balancing the *How* of Digital Tools with the *Why* When Designing Technology-Heavy Writing Courses

Dawn S. Opel, Michigan State University
Mike McLeod, Drawbridge, Inc./Eli Review

This article extends the conversation from the 2012 Computers and Writing conference town hall "Program or be Programmed: Do we Need Computational Literacy in C&W?" by discussing the pedagogy of computation through a media archaeology of an advanced web authoring course in an undergraduate rhetoric and writing program. Designed in 2008, and taught again with the same design in 2016, two instructors of the course reveal, through its origin story and legacy, the cultural and material conditions inherent in the teaching of web design and/or development in a writing program. We stress the importance of the balance of functional, critical, and rhetorical literacies when teaching courses centered on the building of new or emergent technologies. The designer of the course offers the foundational concepts for the course as well as practical suggestions for the teaching of advanced web design. The media archaeology of this course, old and new, suggests ways to avoid potential traps that a teacher may fall into when teaching a new tool at the expense of focus on theoretical concepts that remain constant across technologies and time periods.

In the 2012 Computers and Writing (C&W) conference town hall "Program or be Programmed: Do we Need Computational Literacy in C&W?" Sample and Vee (2012) introduced the role of code in writing studies through not only research but also pedagogy, noting several scholar-teachers who had offered courses in rhetoric and writing that involved a coding component. The town hall explored the degree to which students of rhetoric and writing should be exposed to code: should students learn to code, and what level of technical proficiency should they achieve in order to do rhetorically and culturally meaningful work? Balancing functional, critical, and rhetorical literacies in emergent technology is an evergreen consideration in our field. While technologies may constantly evolve (and perhaps the Learn to Code movement represents our current cultural moment around emergent technology), Selber's (2004) *Multiliteracies for a Digital Age* provides an ever-applicable conceptual framework for balancing technical, critical, and rhetorical skill building for course design involving computer literacy.

This article extends the conversation of computational literacy in the C&W classroom by focusing on an advanced web authoring course in a rhetoric and writing undergraduate program. In this context, the question of whether students should learn to code is already decided—if they have chosen to take this sequence of courses, they will learn. However, the notion of what is advanced in web design in a rhetoric and writing program is in and of itself a nuanced one; here, a division was created between basic web design, which focuses on building static web pages in HTML and CSS, and advanced web design, which focuses on building more dynamic and interactive web applications. This division might at first glance seemingly reflect a commitment to the development of functional and computational skill building, but, as we will discuss, programming dynamic and interactive applications necessitates a rhetorical understanding of the web that cannot readily be understood without hands-on experience. In this way, the evolution of the web from primarily static content to dynamic, interactive content represents another of what Delagrange (2011) refers to as a "*punctum* of technological change" where

> the practices and habits of mind associated with old media are called into question as we struggle to devise principles and practices for the new. What makes these questions important to ask *again* is that points of remediation have in the past been both opportunities for change and occasions for re-inscription of previous practices (*v*).

To that end, we offer an archaeology of sorts of the Advanced Web Authoring course in this program. We say "of sorts" because we are likening this course to a new and emerging media object unto its own, one for which a close examination of its past may offer some insight into the ways in which the technologies that came before it and continue to emerge may influence the making of media in the present

and future (Parikka, 2012). To conduct this media archeology, co-author Mike begins with the origin story of the course, including what practices and habits associated with old media—namely, focusing on the teaching of specific tools such as Flash and Photoshop—caused difficulties in shaping the rhetorical understanding of advanced web design. Mike outlined his re-design of the course in 2008. Co-author Dawn picked up several years later, revisited the curriculum in 2016 to incorporate advancement in web technologies but remained true to Selber's framework and the balancing work in the earlier iteration of the curriculum.

This archaeology is meant to provide useful artifacts as well as critical analysis of the pedagogy of teaching technology. We include teaching materials so that C&W teachers might use what proves helpful to them in teaching computational literacy in the writing classroom. It is meant to show how the past informs the future in the ways that Delagrange as well as media archaeologists and historians such as Parikka (2012) and Gitelman (2006) discussed. There are potential traps that a teacher of technology may fall into, the most serious of which is being enthralled by our technologies, in getting lost in the wonder and excitement of learning a new tool at the expense of understanding a theoretical concept that remains constant across technologies. For example, while English grammar and syntax would not be considered enthralling in a composition classroom today, when grammar is considered in a language like JavaScript, it runs the risk of becoming so. For this reason, we ground our takeaways in histories of old and new media, so as to reveal what is "always already new" about the web (Gitelman, 2006), and, more important for us as teacher-scholars, the way we teach writers for the web.

Origins and Legacy of "Advanced Web Authoring"

Mike was hired to teach Writing, Rhetoric, and American Cultures (hereafter, WRA) 410: Advanced Web Authoring (410), for the 2008 spring semester at Michigan State University. It was a companion course to WRA 210: Introduction to Web Authoring (210); 210 was a required course for all students in the Professional Writing (PW) program while 410 was an elective. While 410 was not new, it wasn't yet cogent. It was named web authoring and not design because of a disciplinary dispute over design ownership. The vagueness of the word authoring resulted in a large scope. The official course description at the time was: "WRA 410 Advanced Web Authoring (3 credits). Developing and maintaining large-scale, interactive Web sites. Visual design, usability, audio and video integration, ongoing site management, and web accessibility."

While reviewing materials from previous instructors, Mike was shocked by how much of the curriculum was consumed by the audio and video" components of that description. At least one-third of the course was dedicated to teaching visual rhetoric using Adobe Photoshop and Flash. While there is certainly merit in that subject and those tools, including them in this course was redundant (PW had since added a required course in visual rhetoric) and the time invested in teaching those tools drastically cut into the time available for the fundamentals of designing for the web. The impact of this focus on Adobe tools became deeply felt when asking prior 410 students why a professional writer might be asked to take a web design course, or what business they would have applying for a web design job, and they were hard-pressed to answer.

Dawn came to 410 in Fall 2016, and discovered that in the years since Mike departed, many of the same cultural concerns still plagued the course: the disciplinary considerations of where web design and web development were taught on campus and the course description outlining functional literacy rather than critical or rhetorical literacy (Selber, 2004). While Flash and Photoshop as particular tools had not been taught in the course in years, a general confusion still existed over how this course differed from a computer science course. A critical reflection of the emphasis on audio and visual components also revealed those aspects of multimodality in which rhetoric and writing faculty had more familiarity—the further the course moved toward web design and development, the more difficult it was to staff and to build confidence in the expertise of the teacher. However, as Shipka (2011) noted, this is a common fallacy in teaching multimodal courses: the teacher is the arbiter of the effectiveness of texts produced. This notion privileges the functional literacy skills of students and places the teacher in the role of expert

of technology. Mike's course design (below) addressed this in its assessment strategy, but the lore surrounding web design and development continued to invoke cultural stigma for both students and teachers.

The Design of Advanced Web Authoring

Mike had two primary goals in designing a new curriculum for Advanced Web Authoring. First, he wanted to build clear bridges between related concepts covered in other program courses (particularly visual and digital rhetoric), making sure students understood those connections but leaving more room in 410 for foundational concepts. Second, he wanted to make sure that the course was foregrounded in concept and theory rather than technology. Where previous sections of the course had invested in learning specific tools or platforms, we would spend no time using tools like Photoshop, Flash, or Dreamweaver. Focusing instead on foundational principles and writing code by hand, students would build a set of transferable skills.

For this, Mike identified three core concepts for 410 that connected web design to writing theory. He developed a project around each concept, utilizing readings from industry sources and tutorials. He also identified one scholarly piece from technical communication scholars for each concept that would help bridge the technical work to rhetorical theory. He focused assessment on design genres rather than markup or code. This not only helped to expose students to the types of writing they might be expected to produce as part of a web design team, but also provided opportunities to reflect on foundational concept attainment and whether they were in the right mindset before they started building their projects. Rather than solely assess the finished code (which could be plagiarized), Mike graded primarily writing they did *about* their technical work.

Concept 1: Separation of Content from Presentation

Despite being an advanced course, many students come to 410 years after 210, requiring a refresher of some basic concepts. The separation of content (HTML structure) from presentation (CSS) is critical for writers, not just developers. Many will be expected to produce content as *writers* and apply structural markup to it (when writing in/for a content management system, for example), while others may work as *designers* to style and arrange those structured texts for multiple audiences. A thorough understanding of the Document Object Model (DOM) makes it possible for a writer not only to create texts for the web, but also create multiple experiences of that same text depending on the needs of a user.

Reading. To reinforce that, students read William Hart-Davidson's (2005) "Shaping Texts that Transform: Toward a Rhetoric of Objects, Relationships and Views." The text is approachable for an undergraduate student and explains how designers create documents that are able to transform to meet the needs of multiple audiences and purposes. These rhetorical strategies involve both ancient practices and contemporary, such as in the design of object-oriented user interfaces. Hart-Davidson laid out "five qualities of valuable web content" that offer a clear value proposition for why writers should, for example, create list items using valid HTML tags rather than hitting the spacebar repeatedly to simulate indentation or relationships (2005, 29).

Project. Students design and execute a fully functional static website, built only from HTML and CSS. Mike encourages them to consider developing their senior portfolio websites, which are a requirement of their major, but the students can choose the content. Along the way, they develop a series of deliverables that eventually add up to a full site.

- ***Planning***. Students establish designs before building using genres like wireframes and design comps; this way they learn the conventions of these genres and also have opportunities for early feedback.
- ***Building***. Students then execute their designs, again with many opportunities to get feedback on their document markup and on their CSS. They are evaluated on how effectively they are

able to structure an HTML document and apply styles to that document and demonstrate an effective separation of both.

Learning Indicators.

- Wireframes demonstrate how well students understand document objects separated from design: wireframes are meant to communicate structure, not style, and readiness to build standards-compliant document markup.
- Their design comps demonstrate how they are thinking about the application of designs in their wireframes and, again, whether or not they understand the application of type, color, etc., to the document structure.
- The HTML and CSS they develop based on their designs are the ultimate indicators of if they successfully accomplished their design goals. W3C validation is required, and then Mike manually inspects their markup to see just how effectively they separated their content from presentation.

The instructor shares the project rubric with students on the first day of class so they can see how everything from the first two weeks builds to the completed project (the portfolio). Breaking it up in this way helps students understand not only the process but also final assessment. It also helps alleviate concerns about mastery of the technology: seeing the project from a higher level to understand that it is much less about learning the intricacies of the technology and more about the concepts as demonstrated in the planning documents.

Concept 2: Information Architecture and Writing for/with Databases

Students read "Shaping Texts That Transform" in the first unit not only because it reinforces why writers should care about markup language, CSS, and the DOM, but because it also introduces a second level of abstraction involved in writing and designing for the web. Most of the modern web is driven by databases, or systems that store information and repurpose that information in a variety of ways. This is a not just a separation of content from presentation, but a separation of content from *structure*. A content management system, for example, allows storage of writing separate of layout, making it possible to drop content into any template with any structure. This allows a service like Amazon to display its catalog content both on their website *and* in a mobile app *and* in third-party services like sites that track pricing trends. In many cases, writing for the web is not about designing content for a specific audience or a specific display, but designing content that can be repurposed for any number of audiences and situations. Writers need to understand this conceptually if they are going to be well prepared to write and design for the web. They experience this daily in every web-based service they use, but most have not engaged with it in a conscious way.

Reading. Students read Barbara Mirel's 1996 chapter "Writing and Database Technology: Extending the Definition of Writing in the Workplace." It was published in the earliest days of the World Wide Web and does not describe working with the web at all, but it provides great insight into the rhetorical processes behind using database technologies to develop reports for a wide range of audiences and purposes. Not explicitly connecting to the web makes this piece even more valuable—highlighting the rhetoric behind database-driven composition is much easier when students themselves have to make the connection between technologies.

Project. Students design and build a simple cataloging system that would focus on one type of object (most students chose to design a system to catalog their book or movie collections, others focused on objects like recipes or guitar tabs). They design and build a database to store information about their object (they entered multiple records, or data about particular representations of their object) and then build a series of displays for their object's records.

The goal of this project is not to have the students build something as complex as a content management system, but to give them experience working with databases and using granular database content to create custom views and displays for different audiences and purposes. The project also provides a very focused and guided first step into server-side programming for the web—their experience

to this point is client-side (markup and design for the browser)—but, to create and use database content, they need to do some programming.

Mike adopted PHP and MySQL to create this project not because these are the best technologies but because they are the most widely used and in tools students would likely be expected to use in the future (e.g., WordPress). Extensive resources are also available online for students who want to learn more or need additional support beyond the classroom.

Learning Indicators.

- Students first produce database diagrams that detailed the types of metadata they would use to describe their objects and serve as a blueprint for the database tables they would actually build. These diagrams show whether or not the student understands their object as an *information* object and whether or not they will collect all of the information they need to create an effective project.
- Wireframes for their catalog show if they have grasped the scope of the assignment (that they know they need to create multiple views of their objects) but also whether or not they have mapped their design to content from the database they built - each piece of information on a wireframe has to be sourced to their database content, and the wireframes show if students grasp that.
- A functional project: A working database-driven website demonstrates that a student grasped both levels of abstraction (separation of content from presentation as well as content from structure) and ability to build context-specific views using granular bits of content from a database. The website also shows that the student built on concepts from the first unit, and the code for those documents shows how they were building their database queries and utilizing the bits.

Students receive the Unit 2 rubric on the first day, again so they have a sense of scope of the project and of how Mike assesses how well they grasped the concepts. This is another opportunity to put the students at ease about working with a new technology—the rubric heavily emphasizes the planning genres, with less weight on the technology.

Concept 3: Writing with/for Content Management Systems

While learning Concepts 1 and 2, students get a taste of the kinds of writing and design they be expected to do most often on the web. The last concept of the course was to experience exactly what "writing" often looks like on the web by introducing them to content management systems. The PW program already had a dedicated content management course, but that course was heavily focused on concepts and organizational culture and less on the nuts and bolts of setting up, writing in, or maintaining a content management system (CMS).

There are arguments to be made for many open-source content management systems, but Mike chose WordPress for its ubiquity. WordPress is an open-source platform with extensive documentation, third-party plugins for customization, and large communities where users can find support. WordPress currently drives more than a quarter of the entire web, including major content organizations like Conde Nast (Protalinski, 2015). Odds are that professional writing students will not only be expected to write for an organization that uses WordPress but that they might also be asked to install, customize, or maintain it.

Reading. Mike chose Dave Clark's (2008) "Content Management and the Separation of Presentation and Content" because it not only explains content management as a concept but also bridges to core rhetorical concepts about audience and customization raised by the earlier readings. It makes a clear case for how writing changes in these systems and discusses the new challenges faced when we no longer write "documents" but rather bits that must be stored, organized, and customized for context.

Projects. Once students have installed and configured WordPress on their own servers, they are tasked with developing a theme and a custom content type. Themes in WordPress are custom templates that seamlessly render the functionality and appearance of sites, such that a user can quickly choose a theme with their intended purpose and audience in mind (for instance, choosing a blog or a portfolio

theme). Building a custom template forces students to engage with WordPress's information architecture and to build new views using their data objects. Likewise, a custom post type in WordPress allows students to expand the types of content WordPress can store, making it possible for WordPress to manage content beyond merely pages or posts, to Frequently Asked Questions, Events, Recipes, and more. Again, students apply what they already know about information design to diagram their objects (most re-use what they designed for project 2) but implemented inside WordPress's garden.

Learning Indicators.

- Design documents: As with other projects, students plan their products using wireframes and database diagrams and use those for early feedback before developing any code.
- Themes: Themes in particular demonstrate that students understand WordPress's information architecture. Developing a theme requires them to know how to work with WordPress's content engine known as The Loop and use those granular bits (title, author, publish date, body, tags, etc.) to create new layout and organization.
- Custom Post Types: Here, again, learning can be assessed early. For a custom post type, students must develop a form that lets them create new entries and then a new template for displaying that content. Each moment is a place to stop and assess whether they have executed the concepts from their designs effectively and to intervene when necessary.

While still very important, design genres are not as prevalent in Unit 3 projects as in earlier units because, at this point, students will have considerable experience writing in these genres. This unit relies on feedback groups much more extensively, sometimes having students meet at each session to share their progress or give input on their materials. Students are usually comfortable and confident enough by this point to give each other helpful feedback and help each other recover when necessary, giving them more opportunities to engage with and reinforce the concepts.

Conclusion: Implications for 2016 and Beyond

When Dawn began teaching 410 in fall 2016, Mike shared his course materials with her (he had not taught the course for several years), and she began the process of updating the technical specifications of the course to catch up with current web standards. However, Dawn wanted to emphasize that updating functional technical components of the course (for example, moving to the latest web standards: XHTML to HTML5 and CSS to CSS3) was not difficult because course's focus on a rhetorical understanding of the web. Dawn also added a final unit (that she discussed in this Sweetland DRC blog post) to connect all course concepts through a collaborative social networking site re-build. But as Mike mentioned above, the rhetorical theory that is the backbone of the course remained the same. While technological advancements will continue to accumulate, the goal of the course is to ensure that, conceptually, students understand what the web is and why they as professional writers might design and write for it.

In Dawn's blog post, she attempted to connect critical making inherent multimodal composition pedagogy (see Shipka, 2011) to web design and development, particularly as the latter are brought into PW and rhetoric and writing courses' curricula. While we focus here on an advanced web design course, it is crucial to mention that the concepts here can be translated to any course in which digital tools are used. While this course focuses on the tools of web designers and developers, educational technology tools, graphic design tools, and many more are now embraced in rhetoric and writing courses. The scaffolding we present above for each project (a concept, a reading focused in rhetorical theory, a project with applied and iterative components, and learning indicators that reflect both rhetorical and technical knowledge) can be used with any technology. For programs interested in adding web design to their PW or rhetoric and writing programs, a good first step might be to compare the pedagogy for the teaching of tools in other multimedia courses in the pre-existing curriculum. Aligning these creates consistency and eases students into learning new and emergent technologies across contexts.

A final important implication to this "media archaeology" of 410 is situated in the material, cultural, and political nature of making, expertise, and assessment in a web design course in a PW program. If, as the course description describes, developing and maintaining websites is the primary task of the course,

void of context, it situates functional technological literacy as the sole goal of the course. This has two negative effects. First, it terrifies both writing teacher and student alike as they imagine memorizing syntax and taking (and grading) timed coding tests. This fear existed in 2008, and this fear will exist in 2108, unless the rhetorical, the why, is foregrounded in our discussions of web design in C&W and in rhetoric and writing more broadly. Second, when functional technological literacy is foregrounded in curriculum design, both teacher and student perceive expertise to be situated in the ability to code, rather than in why it might be important to know *how* to code as well as *why* and *for what world* we are creating with that skillset. The power of a web design course in a PW program is the possibilities that exist when a critical maker enters the workforce armed with the desire to do good. An archaeology of this course reveals that this student-as-critical maker can be supported by web design curriculum that encourages rhetorical awareness and critical questioning of current industry tools.

References

Clark, Dave (2008). Content management and the separation of presentation and content. *Technical Communication Quarterly, 17*(1), 35–60.

Delagrange, Susan H. (2011). *Technologies of wonder: Rhetorical practice in a digital world*. Logan, UT: Utah State University Press. Retrieved March 1, 2017, from http://ccdigitalpress.org/wonder/

Gitelman, Lisa. (2006). *Always already new: Media, history, and the data of culture.* Cambridge, MA: MIT Press.

Hart-Davidson, Bill. (2005). Shaping texts that transform: Toward a rhetoric of objects, relationships and views. In Carol Lipson & Michael Day (Eds.), *Technical communication and the World Wide Web* (pp. 27–42). Mahwah, NJ: Lawrence Erlbaum.

Mirel, Barbara. (1996). Writing and database technology: Extending the definition of writing in the workplace. In Patricia Sullivan & Jenni Dautermann (Eds.), *Electronic literacies in the workplace: Technologies of writing* (pp. 91–112). Urbana, IL: National Council of Teachers of English.

Parikka, Jussi. (2012). *What is media archaeology?* Malden, MA: Polity Press.

Protalinski, Emil. (2015). WordPress now powers 25% of the web. *VentureBeat*. Retrieved March 1, 2017, from https://venturebeat.com/2015/11/08/wordpress-now-powers-25-of-the-web/

Sample, Mark, & Vee, Annette. (2012). Introduction to the role of computational literacy in computers and writing. *Enculturation*. Retrieved March 1, 2017, from http://enculturation.net/computational-literacy

Selber, Stuart. (2004). *Multiliteracies for a digital age*. Carbondale, IL: Southern Illinois University Press.

Shipka, Jody. (2011). *Toward a composition made whole*. Pittsburgh, PA: University of Pittsburgh Press.