Seeing the Code: Text, Markup, and Digital Humanities Pedagogy

Dissertation

Presented in Partial Fulfillment of the Requirements for the Degree Doctor of Philosophy in the Graduate School of The Ohio State University

By

Trey Conatser, MA

Graduate Program in English

The Ohio State University

2022

Dissertation Committee:

Scott L. DeWitt, Adviser

Jonathan Buehl

John Jones

Ben McCorkle

Abstract

What is the value of code in the humanities class, and what does it do for a humanities education? To what degree does code help us think about and compose texts, and to what degree can we engage with it as a text itself? Guided by these framing questions, this dissertation lies at the nexus of digital humanities; rhetoric, writing, and composition; and teaching, learning and pedagogy. It engages coding as a fixation of the global information economy: a literacy that has joined reading and writing to constitute a foundation of "moral goodness and economic success" signaling "the health of a nation and its citizens" (Vee 3).

The larger argument of this dissertation is developed around the notion of *seeing the code* as a pedagogical framework for teaching and learning with code in the humanities. Scholars have begun to investigate how we can think about code and coding cultures vis-a-vis literacy studies, rhetoric, and the hermeneutical methodologies of the humanities. This dissertation extends the developing humanities framework for analyzing and composing with code into the larger discourse on teaching and learning with code. Just as the past few decades have seen the multimodal turn in writing and humanities pedagogy, this dissertation looks ahead to a coding turn that will just as much naturalize a peculiar medium of representation and agency as part of the teaching mission of our disciplines.

The overall goal of the dissertation is to construct a rigorous, multidimensional, and transdisciplinary ethos for digital humanities pedagogy—and code-focused pedagogy in particular—that draws from research and teaching in rhetoric, writing, and textual studies; the (digital) humanities broadly; education studies; and science and technology studies. Chapter one

develops a vernacular theory of code by calling on a variety of phenomena and disciplines. I examine how code resonates with and advances learning goals in the humanities, particularly for rhetoric, writing, composition, and textual studies. Chapter two traces the debate on coding in digital humanities scholarship from 'the digital humanities moment' in the early 2010s, which raised new questions about what code can and should be doing for humanities scholarship, how code allows us to think about issues both perennial and unprecedented, and what the evolving nature of code signals for the future of the digital humanities community.

The final two chapters examine extended pedagogical case studies on the use of markup language as a particular form and genre of code. Chapter three investigates the curricular and pedagogical designs behind the use of extensible markup language (XML) in the first-year writing class, and chapter four focuses on an advanced undergraduate seminar on archival research and digital scholarly editing. These case studies share an investment in leveraging markup language to render students' compositional and/or conceptual moves visible with the goal of developing more critical and reflective learners. All in all, code provides an opportunity to advance the humanities' signature pedagogies, methodologies, and scholarship with respect to a novel, powerful, and increasingly ubiquitous form of technology and communication.

Dedicated to Ruthie and Carly

Acknowledgments

This dissertation emerged from a process that involved coming to terms with the need to shift my area of specialization once I was well into an initial project focusing on British literary history. Despite the commonsense wisdom that finishing as quickly as possible represents the most desirable outcome for a graduate student, I recognized that my academic and professional journeys were inexorably changing. Graduate positions involving rhetoric, writing, composition, digital studies, and, importantly, teacher and educational development led to scholarly projects in digital humanities and pedagogy, culminating in a full-time position in a center for teaching and learning at a state flagship research university where I've taken on leadership roles in service to the institution's teaching mission. Precisely because of this work and milieu I've come to view my dissertation project and progress quite differently; I offer it here as the culmination of a wide range of experiences, expertise, and exigencies with the goal of providing both conceptual and concrete guidance for innovative teaching and learning with code in the humanities.

I am grateful for the singular opportunity I had at Ohio State to work directly with Cindy Selfe and Scott DeWitt as part of the storied Digital Media and Composition (DMAC) Institute. Their mentorship and collaboration were a curriculum unto itself that, first and foremost, led me to discover an affinity and aptitude for digital media studies, writing and composition, pedagogy, and educational development with which I otherwise would not have credited myself. My work with them connected me to a network of scholars, educators, and colleagues who continue to energize and inspire from both near and afar. I do not take for granted the good luck I had to be in the same department as Cindy and Scott, who so generously took me under their wings. I am

further grateful to Scott for agreeing to chair my reconstituted committee despite the unique situation in which I found myself, balancing a full-time academic staff role in a different state while researching, writing, and making progress when I could. With kindness, patience, and clearheaded guidance, Scott helped me navigate the challenges of the project, particularly as they related to aligning it with the changing nature of my scholarly and professional situation. I may never have appeared on Cindy and Scott's radar were it not for my colleagues in the program, Krista Kurlinkus and Katie DeLuca, who encouraged me to pursue positions they had previously held in the English department's Digital Media Project as well as the DMAC Institute.

I very much appreciate the willingness and generosity of my committee members—Jonathan Buehl, John Jones, and Ben McCorkle—to serve and provide encouragement and feedback. None of them had any obligation to serve on the committee, and they helped to get me across the finish line at the end of a long journey. This dissertation may never have happened were it not for Louie Ulman's seminar in electronic textual editing during my first quarter at Ohio State. I had never encountered XML, the TEI, etc., nor considered how markup language and code might have a place in our conversations about humanities scholarship and education. Louis clearly saw my interest in the work and was generous to include me in future opportunities for collaboration and presentation. Eddie Singleton, director of first-year writing at the time, did not hesitate to green-light what just as easily could have appeared as a harebrained proposal to take an already busy curriculum and integrate coding into all of it. The class remains one of the highlights of my career and the critical response to my reflections on teaching it was, frankly, surprising. From invitations to publish more formally to seeing my work appear in graduate

seminar syllabi and doctoral dissertations, this response conveyed that there may be something significant behind what otherwise could have remained a one-and-done experiment.

Kathleen Griffin is a hero of the English department at Ohio State, and I would not have been able to complete this project nor my degree program without her wisdom in navigating both departmental and institutional processes—and cultures—with such a genuine interest in my success over the years. As director of graduate studies, Aman Garcha offered much needed momentum and support for my project especially at the critical moments when I was adapting to a new home, job, and life; it was a crucial tether to know that my department was still behind me and my work. Before my shift to digital humanities and educational development, I learned and benefitted tremendously from the expertise and warm community of world-class scholars in the English department: Sandra Macpherson, Roxann Wheeler, David Brewer, Alan Farmer, Les Tannenbaum, Jill Galvan, David Riede, and Jake Risinger, among others. My friends and peer-colleagues in the department are too numerous to list here, and they all now enjoy success within and beyond the academy. Thank you all for the company and camaraderie that were essential not only for my wellbeing but also for the sharpening of my intellect among such energetic and incisive minds. I am lastly (though not least!) grateful to Clare Simmons, who served as my adviser for nineteenth-century British literary history during my funded time in the program. Clare's guidance got me to many important milestones, and her help was just as critical as she recognized the sea changes in my scholarly interests and helped to make that shift as easy and empowering as possible.

Though I submit this dissertation at Ohio State, my teachers and mentors from other institutions have been just as critical for the intellectual development that led to this project.

Marco Abel, Hilda Raz, Grace Bauer, Steve Behrendt, and Ken Price, among others at the University of Nebraska-Lincoln, welcomed me into English studies as a graduate student and got me started on the scholarly apprenticeship that culminates with this document. In Steve and Ken's classes I had my first encounters with digital humanities work, and Marco's commitment to professionalizing graduate students from day one has paid dividends long after I took his seminar. I credit much of what I've been able to accomplish to my ability to use language with clarity of meaning without sacrificing the craft, style, and voice that are so critical to persuasion. Hilda and Grace's mentorship in creative writing and poetry rewired my brain when it comes to words and language; my ability to exert agency in written and spoken forms has benefitted tremendously as a result. A conviction that I carry to this day is that all writing is creative.

Before my MA program I was something of a dilettante in my undergraduate education, moving from music performance to architecture before finally arriving at English studies for no reason other than a short-lived attempt to write a fantasy novel. After graduating high school, in fact, I had somewhat celebrated that I would never again take an English class because I had tested out of the requirements for Texas universities at the time. However, Tim Richardson's courses in literary criticism and poetry were eye-opening for what they revealed about the rigorous, creative, and *important* work being done in the field. Tim became an informal adviser of sorts, and his encouragement to apply for graduate school was significant at a time when I was weighing my options for post-baccalaureate life. I enjoyed a small but tightly knit community of peers at the University of Texas-Arlington, and I am particularly grateful for the friendship and humor of Tony Burns, with whom I keep up to this day as he too fights the good fight for English studies and humanities education.

Much later, when I was once again weighing my options at a crossroads, I saw a job announcement come through the WPA-L listserv for something called a faculty instructional consultant at the University of Kentucky's Center for the Enhancement of Learning and Teaching (CELT). I hadn't before worked with a teaching center—though was aware of what then was called UCAT at Ohio State—but the job seemed fortuitously aligned with the work I had done for the DMAC Institute. I knew I was lucky to end up with the job offer, but it was only later that I knew how lucky I was given that I had been up against an intimidating pile of applicants with impressive credentials and accomplishments. I owe many aspects of my career to Kathi Kern, the founding director of CELT and later Associate Provost at UK, who picked me out of the pile and had nothing but confidence in my ability to grow into the role. In the years since, Kathi has become a dear friend and trusted adviser, even as she now lives several states away. Because of her mentorship and generosity in involving me in so many aspects of the work, I have received a singular education in academic administration and leadership, where one foot stands firmly in the world of faculty, academics, teaching, learning, and scholarship, while the other stands among executive administration and institutional operations. More than this, Kathi unwaveringly encouraged me to continue chipping away at the dissertation project and offered critical advice too many times to count. Thank you, Kathi, for believing in me.

Also at UK, Phil Harling has been a source of kindness and encouragement as he helped to brainstorm approaches when the project seemed stalled. Moreover, I am grateful to Phil for eagerly offering up a section of the honors seminar for what eventually became the basis for the fourth chapter of this dissertation, which has since appeared in a Palgrave handbook on public and digital humanities. My colleagues in UK Libraries have been critical to this aspect of the

project as well, particularly Jennifer Hootman and Sarah Dorpinghaus, who, among other things, formed a team with me that was accepted into an NEH Institute on sustainability for digital humanities projects. I have had many colleagues in my role at CELT, but two who have helped me cleave to my ambitions are Jill Abney and Shauna Morgan. Their friendship and persistence in reminding me of what's possible have kept me going (and working on my project) especially when scopes of responsibility have bloated in the post-pandemic academy. At UK I have had the good fortune to develop a network of colleagues from all disciplines and levels of academic work; I truly feel like I belong here and will be eager to take the momentum from defending this dissertation into the work I do every day for the benefit of the UK teaching community.

My family has always encouraged my learning and academic success, even if they didn't quite understand what I was doing or why I was doing it. I am grateful to my mother, father, Tammy, and Vance for supporting me both materially and symbolically throughout my years as a wandering academic (who now has finally settled down, I hope). I met my wife because of the DMAC Institute, and she has always insisted that we could find a way to make things work so I could get the space and time to finish this project. Especially after our daughter was born, Carly has happily taken on solo-parenting duties so that I could steal away during the weekends and work at a coffee shop or hunker down in the home office during weekday evenings. The logistics of graduate programs are not easy for students with jobs or caregiving responsibilities; I could not have done this without Carly's labor, help, encouragement, endless listening, and, of course, her love. Finally, while she does not yet understand what a dissertation or degree is, I am forever grateful to my daughter Ruthie for the meaning she has given my life in all its facets, including the renewed drive to make the most of my opportunities. Whenever I get too caught up in the

grown-up world, spending time or playing with Ruthie reminds me of the joy we can take in the moment, in the company of family and loved ones. Perhaps one day she'll see this dissertation and we can reflect on the impact it will have had on my development and career. Perhaps at that point she will have already done some coding herself.

Vita

May 2006……B.A. English, University of Texas at Arlington

May 2008……M.A. English, University of Nebraska-Lincoln

Aug 2008 to May 2011……...Instructor, Ivy Tech Community College of Indiana, Indianapolis

Sep 2011 to Aug 2012……Graduate University Fellow, Department of English, The Ohio State University

Aug 2012 to May 2016……...Graduate Teaching Assistant, Department of English, the Digital Media Project, and the Center for the Study and Teaching of Writing, The Ohio State University

Sep 2013 to July 2015……Graduate Associate Director, Digital Media and Composition Institute, The Ohio State University

June 2016 to June 2018……...Faculty Instructional Consultant, Center for the Enhancement of Learning and Teaching, University of Kentucky

Aug 2016 to May 2019……...Part-Time Instructor, The Graduate School, Department of Rhetoric, Writing, and Digital Studies, and the Lewis Honors College, University of Kentucky

July 2018 to Feb 2022…….Associate Director, Center for the Enhancement of Learning and Teaching, University of Kentucky

March 2022 to present……Acting Director, Center for the Enhancement of Learning and Teaching, University of Kentucky

Publications

Conatser, Trey. "Open Pedagogy and the Archives: Engaging Students in Public Digital Humanities," forthcoming in the *Palgrave Handbook of Digital and Public Humanities*, edited by Tara Thomson and Anne Schwan. Palgrave Macmillan, 2022, pp. 237-256.

Carman Angela, Christie Brady, and **Trey Conatser**, "Game Design for High-Order Thinking in Review Activities." *College Teaching*, vol. 70, no. 1, 2022, pp. 119-123.

Conatser, Trey, et al., "Coding (and) the Archive: Texts, Markup, and Open Pedagogy." *Midwest Archives Conference Newsletter*, vol. 47, no. 1, 2019, pp. 29-31.

Cardarelli, Kathryn, Angela Carman, and **Trey Conatser**, "Engaging the Public Health Student through Active and Collaborative Learning," in *Teaching Public Health*, edited by Lisa M. Sullivan and Sandro Galea, The Johns Hopkins University Press, 2019, pp. 147-157.

Conatser, Trey. "Keeping Track of DMAC: Visualizing Influence Across Space and Time." *Computers & Composition Online*, 2015, cconlinejournal.org/ciwic_dmac/conatser/.

Conatser, Trey (ed.). *Showcasing the Best of CIWIC/DMAC: Approaches to Teaching and Learning in Digital Environments*, The Digital Media and Composition Institute, 2015, www.dmacinstitute.com/showcase/issues/no1.

Conatser, Trey. "Changing Medium, Transforming Composition." *Journal of Digital Humanities* vol. 2, no. 2, 2013, journalofdigitalhumanities.org/2-2/changing-medium-transforming-composition-by-trey-conatser/.

Conatser, Trey. "Hawk Poem," "What the Corpse Said," "The Heads of the Apostles at Notre Dame." *Prairie Schooner*, vol. 86, no. 3, 2012, pp. 133-137.

Conatser, Trey. "To Be(head) a Family: British Poetry and the Reclamation of Marie Antoinette." *CEA Critic*, vol. 72, no. 2, 2010, pp. 1-22.

Conatser, Trey. "There's No 'I' in Information: Some Naysayings for New Media Studies." *New Media & Society*, vol. 12, no. 3, 2010, pp. 365-378.

Conatser, Trey. "The Aphasia Epidemic," "The Ding Dong Ditch Epidemic." *Diagram*, vol. 10, no. 4, 2010, thediagram.com/10_4/conatser.html.

Conatser, Trey. Review of *Dog Years* and *Fire to Fire: New and Selected Poems*, *Prairie Schooner*, vol. 83, no. 2, 2009, pp. 195-199.

Conatser, Trey. "A I," "Permutations on a Horse." *Copper Nickel*, vol. 12, 2009, pp. 74-79.

Conatser, Trey. "For the Not-Yet." *Portland Review*, vol. 56, no. 1, 2009, pp. 104-105.

Conatser, Trey. "Goldilocks." *Mid-American Review*, vol. 29, no. 2, 2009, pp. 9-10.

Conatser, Trey. "I'm not Steve Brodie." *Seneca Review*, vol. 39, no. 1, 2009, pp. 35-38.

Conatser, Trey. "Leap, frog." *New York Quarterly*, vol. 65, 2009, p. 96.

Conatser, Trey. "Impact." *South Dakota Review*, vol. 46, no. 3, 2008, pp. 32-33.

Fields of Study

Major Field: English

Minor Field: Digital Humanities

Minor Field: Composition and Writing Studies

Minor Field: Teaching, Learning, and Pedagogy

Table of Contents

List of Figures

Introduction

"[W]hen we don't know how code works…we make unknowingly, no matter how skilled we might be working through a program's user interface."

Stephen Quigley, "Basic Coding"

Unfortunately, outside of specific circles, these various conversations on digital media—whether education-oriented or not—tend to focus on software as an instrumental tool and thus ignore or otherwise fail to address the role that meaning making, and in particular meaning made in and around code, *plays in the development and use of software. Code, however, can also be approached rhetorically and critically in reflection of its meaningful nature.*

Kevin Brock, *Rhetorical Code Studies*

We should be uncomfortable with the ways that raising up the objects of computation risks making the humans creating, using, or feeling their effects invisible.

Brandee Easter, "Fully Human, Fully Machine"

What is the value of code in the humanities class, and what does it do for a humanities education? To what degree does code help us think about and compose texts, and to what degree can we engage with it as a text itself? Guided by these framing questions, this dissertation lies at the nexus of digital humanities; rhetoric, writing, and composition; and teaching, learning and pedagogy. It engages coding as a fixation of the global information economy: a literacy that has joined reading and writing to constitute a foundation of "moral goodness and economic success" signaling "the health of a nation and its citizens" (Vee 3). In a public address aimed at students during Computer Science Education Week in 2013, President Barack Obama declared that "learning these [programming] skills isn't just important for your future; it's important for our country's future." This decree followed the 2012 Year of Code campaign that also sparked a

robust debate over the value of learning to code and involved publicity stunts like New York City Mayor Mike Bloomberg tweeting a new year's resolution to enroll in Codecademy, an online platform for free coding courses launched in 2011.[1] Overall, the past decade-plus has seen coding literacy and education become something of a land rush as prospectors race to lay claim to an apparent wealth of economic, political, and social capital that, like oil or gold, has been discovered in dramatic fashion on previously unremarkable land. The heightened rhetoric around coding literacy and education has followed larger ideological and organizational shifts in the United States: the perceived value and need for more STEM education, a growing critique of higher education's impact on students' professional and financial prosperity, the rising popularity of alternative and short-term educational credentials, a focus on re-skilling/up-skilling for adult learners and career transitions, dramatic economic shifts and restructuring in the wake of global financial and health crises, the rapid digitization of everyday personal and professional life, and, following that trend, increasing public and academic attention to the promises and perils of a ubiquitously coded world.[2]

Despite lofty ambitions, the drive to universalize coding literacy seems to have reiterated some of the very problems that it purports to remedy. In contrast to his professed enthusiasm for an open-access coding education platform, Bloomberg later expressed class and regional biases for which the information technology sector has long been critiqued. "You're not going to teach a coal miner to code," he remarked during a 2014 energy summit (Smiley). In this case and

[1] In "Responding to the Coding Crisis," Kevin Brooks and Chris Lindgren include a screen capture of Code.org's homepage on Feb. 12, 2013, which featured blurbs on the importance of learning to code by celebrities and politicians from musician will.i.am and NBA star Chris Bosh to President Bill Clinton and Senator Marco Rubio.
[2] See, for example, Shalini Kantayya's documentary *Coded Bias* (2020), which explores the work of MIT computer scientist and social activist Joy Buolamwini, particularly her exposure of racial bias in facial recognition algorithms and software. The documentary was picked up by Netflix in 2021.

others the notion of a universal literacy is operationalized less as a welfare argument (i.e., all deserve access to learning opportunities that lead to heightened rhetoricity and agency) and more as a gatekeeping function drawn from familiar ideologies of class, race, gender, and so on. In 2016 the nonprofit organization Mined Minds began work in West Virginia to great fanfare with the goal of training residents through a coding boot camp, thus spurring economic development among a population whose standing traditionally has been tied to coal. Unfortunately, the effort quickly became a boondoggle with a growing litany of unexpected setbacks, unethical business and teaching practices, and unfulfilled promises. After the collapse of the program and only one participant securing a job in programming, a resident and former student described it as part of a "cycle" by which outside interests intrude into a disenfranchised region with compelling agendas that prove less altruistic and more exploitative, or merely shortsighted and poorly executed, in the long run (Robertson). Writ large, these projects express and inspire a great deal of goodwill —at least in the short term—but, in practice, the goal of teaching coding skills for job readiness or career transitions often falls short in the face of structural barriers and a focus on short-term gains within given economic, social, and political frameworks. Examining coding crisis stories that have circulated over the last decade, Kevin Brooks and Chris Lindgren find parallels with John Trimbur's earlier work on responses to literacy crisis narratives, particularly (1) "a rhetoric of global competitiveness" that overlays "deep-seated cultural anxieties" regarding the United States' place in the world and in world-building, (2) the "individualization" of literacy efforts around credentialing programs aligned with corporate needs rather than "harness[ing] the power of codes or databases" for democratic and social transformation, and (3) "an unarticulated social stratification" that cites "individual, rather than systemic failure" for uneven access to programs

and resources, and that "assimilate[s]" individual differences and diversity into a monoculture. The more ambitious aspirations of coding literacy as it has come to occupy our imagination—the moral goodness, the economic success, the health of a nation and its citizens, the transformative possibilities, and the privileges of rhetorical agency in an increasingly digital and code-driven world—arguably remain unrealized.

Other efforts, however, address coding literacy with systemic, democratic, and justice-oriented aims. Organizations such as Girls Who Code and Black Girls Code seek to instill self-efficacy and eliminate stereotype threat among those traditionally excluded from coding careers and the general IT sector. As Safiya Umoja Noble explains, the value of rectifying the imbalance of representation in the labor of software and programming lies not only in corrective justice but also in the transformation of the digital infrastructures that reify a white supremacist episteme. In the case of search algorithms that privilege certain results and reflect the biases and ideologies of their authors, Noble writes that "women and people of color could benefit tremendously from becoming programmers and building alternative search engines that are less disturbing and that reflect and prioritize a wider range of informational needs and perspectives" (26). Many other educational programs and interventions aim to develop coding literacies, from primary school to upskilling for adult learners, all of which make varying claims on the proposition of coding as an opportunity for economic and social restoration, democratization, or justice. With those claims arise interconnected and deceptively simple questions. Who (and what) is coding for? What kind of meaning can it make? What does it contribute to the individual and community? What are the goals of teaching code in a particular context—especially if that context lies beyond educational programs or disciplines traditionally associated with coding—and how should we facilitate

learning? How do coding literacy and education initiatives bolster, redirect, or challenge extant economic, political, cultural, and ideological forces?

Ultimately, coding itself becomes coded in the sense that we imbue it with meaning and arguments well beyond the technical and self-evident. To what degree, we ask ourselves, does coding—this contested, nascent, yet rapidly growing literacy and digital infrastructure—affect how we (ought to) go about our business in the disciplines and the professions, and how does it affect how we think about our disciplines and professions in the first place? What at first glance may appear to be a niche concern for computer science and the information technology sector reveals itself as a challenge with which we all must contend. As software developer Paul Ford explains to a general audience for Bloomberg's *Business Week*, "Code has been my life, and it has been your life, too. It is time to understand how it all works."

Accordingly, the humanities and higher education have struggled to come to terms with how coding relates to and affects their cultures and goals, particularly with respect to the values, methodologies, subject matter, and learning outcomes traditionally associated with the study of rhetoric, writing, literature, culture, history, media, philosophy, and so on. Yet, coding also seems to present an opportunity to modernize the liberal arts and align the educational enterprise with emergent phenomena that impact our lived experiences: the holy grail—or, depending on one's disposition, the *ignis fatuus*—of 'real-world' relevance for disciplines that lie beyond the strictly professional, technical, and applied areas. Even in digital humanities discourses, coding has occupied a contested position; for some, it represents one kind of work among many options that a digital humanist might pursue while others insist that it's the *sine qua non* of scholarship and production in the field. These valuations and lines in the sand emerge from the questions

with which I began: what is the value of coding in the humanities, and what does it help us do with our methods, subjects, and scholarly goals?

The larger argument of this dissertation is developed around the notion of *seeing the code* as a pedagogical framework for teaching and learning with code in the humanities. Recently, scholars such as Annette Vee, Kevin Brock, and Mark C. Merino have investigated how we can think about code and coding cultures vis-a-vis literacy studies, rhetoric, and the hermeneutical methodologies of the humanities. This dissertation extends the developing humanities framework for analyzing and composing with code into the larger discourse on teaching and learning with code. Much of the scholarly work on coding education features social science and educational studies approaches to assessing the impact of programming and computer science initiatives in a K-12 setting. This dissertation seeks to add more dimensions to that body of work with a focus on higher education and the signature methodologies and pedagogies of rhetoric and writing. As an essay in criticism, seeing the code seeks to define and describe code, to trace its contours and edges, to evaluate its capabilities and limitations, to theorize how it produces and is produced by historical, social, and material forces, to examine the explicit and implicit claims that it makes, to leverage it as a medium, and to deploy it as an intervention in academic and nonacademic settings. Just as the past few decades have seen the multimodal turn in writing and humanities pedagogy, this dissertation looks ahead to a coding turn that will just as much naturalize a peculiar medium of representation and agency as part of the teaching mission of our disciplines.

I also attempt in this dissertation to center pedagogy and add nuance to the discourse on teaching and learning in the digital humanities. A decade ago, Brett Hirsch found that even when digital humanists wrote about pedagogy—and this was in the minority, he emphasizes—they

often did so with a move he calls "bracketing," by which the language of teaching and learning endures "the almost systematic relegation...to the status of afterthought, tacked-on" as a sort of "parenthe[tical]" element (5). We see this bracketing in the opening chapter of the first volume of *Debates in the Digital Humanities*:

> Whatever else it might be, then, the digital humanities today is about a scholarship *(and a pedagogy)* that is publicly visible in ways to which we are generally unaccustomed, a scholarship and pedagogy that are bound up with infrastructure in ways that are deeper and more explicit than we are generally accustomed to, a scholarship and pedagogy that are collaborative and depend on networks of people and that live an active, 24-7 life online. (Kirschenbaum, "What Is Digital Humanities," emphasis added)

Published in the same year as the first *Debates* volume, Hirsch's collection on digital humanities pedagogy seeks to center teaching and learning as an intellectual activity of equal significance to research and scholarship. Since then, scholars have continued to campaign for the value of pedagogical work; for example, Jesse Stommel writes that "pedagogical work should be honored as the best kind of research, and our scholarship should be pedagogical." He considers pedagogy as a kind of technical communication that requires us to "make the work legible" in ways that create new knowledge rather than simply presenting the *fait accompli* of scholarship to novice learners ("Public Digital Humanities" 84). If the digital humanities hold transformational possibilities for how we think about and 'do' our disciplines, how does that transformation happen for *teaching and learning* in our disciplines? In responding to this question, I also keep in mind the critiques of intellectually lightweight and tool-centric discussions of digital humanities pedagogy that dwell on what transpired in courses and learning activities without grounding

them in scholarly history or robust theoretical and/or empirical frameworks (Ball, "Logging On"; Tinnell, "Post-Media"; Earhart). Part of the overall goal of the dissertation, in fact, is to construct a rigorous, multidimensional, and transdisciplinary ethos for digital humanities pedagogy that draws from research and teaching in rhetoric, writing, and textual studies; the (digital) humanities broadly; education studies; and science and technology studies.

Brandee Easter's recent analysis of the perplexing esolang called brainfuck is illustrative of this ethos. Short for 'esoteric language,' esolangs are "obfuscated programming" or "weird languages" that reject traditional notions of clarity, functionality, and sensibility in computer programming in favor of aesthetics, metacommentary, experimentation, avant-gardism, jokes, parody, and so on (Mateas and Montfort 144). As Easter writes, "weird programming languages make arguments *about* code *with* and *in* code" ("Fully Human" 203, emphasis original). She reads brainfuck via queer, feminist, critical race, and postcolonial critiques of mastery—"a difficult, masculine, and material assertion of identity and control"—as a gatekeeping value for the recognition of full personhood, particularly as that personhood is implicitly hegemonic in posthumanist and technocentric discourses (205). The name *brainfuck* itself begs critique for its echoes of sexualized violence and masculine intellectual superiority, but Easter is more interested in how the syntax and semantics of the language both reflect and enact a rhetoric of mastery. Urban Müller designed brainfuck in 1993 with the experimental goal of inventing a coding language that required the smallest possible compiler to execute; the name canonically refers to both the level of difficulty that the language presents and the largely unnecessary and undesirable nature of that difficulty. Eschewing natural language for mathematics and ease of human readability for ease of machine readability, it uses only eight symbols:

>	Move the pointer to the right
<	Move the pointer to the left
+	Increment [the numerical value of] the memory cell at the pointer
-	Decrement [the numerical value of] the memory cell at the pointer
.	Output the character signified by the cell at the pointer
,	Input a character and store it in the cell at the pointer
[	Jump past the matching] if the cell at the pointer is 0
]	Jump back to the matching [if the cell at the pointer is nonzero (“brainfuck”)

As Easter explains, brainfuck “mov[es] a pointer left and right on the memory—a stack or tape divided into cells that can each hold a single value…The machine then works by moving the pointer on the stack, adding or decreasing values at the pointer, and reading or writing stored values” (208).

Easter conducts a close reading of a possible brainfuck rendition of “Hello, World!”, a first-program exercise in any language that outputs the text “Hello, World!” and acclimates a beginning programmer to the coding environment. Because numerical values in brainfuck correspond to ASCII character values, one would need only to include the correct number of + signs at each memory cell to add up to the desired character before outputting it and moving on to the next cell. But this quickly becomes inefficient and does not perform the kind of machinic and minimalistic mastery that the form and spirit of the language calls for. Notably, the code for this basic approach would remain mostly legible as long as the reader understood three of the program’s symbols and had access to the ASCII character values; moreover, it would only require that the reader perform basic counting and follow the sequential logic of spelling in terms of their enactment of computational thinking. Rather, the brainfuck rendition of “Hello, World!” that Easter examines is much more complex and opaque:

```
++++++++++[>+++++++>++++++++++>+++<<<-]>++.>+.+++++++..+++.>++.<<
+++++++++++++++.>.+++.— —.— — –.>+.
```

Breaking the program into three segments, Easter shows how it proceeds by privileging machinic efficiency in ways that do not seem intuitive or legible to a lay reader, e.g., setting up a series of values before outputting characters, looping back and forth through memory cells and changing their values to accomplish the program's goal in the fewest possible steps. As more high-level and sometimes even visually driven programming languages are designed to promote coding literacy by reducing the need to be aware of the abstract and technical aspects of how code actually manipulates computer hardware, "brainfuck poses its challenge through a return to the machine" and even an "intimacy" with machinic 'thinking' by which "programmers prove their (white, male, heterosexual) humanity by becoming indistinguishable from the machine and demonstrating their control, dominance, and mastery while doing so." In short, brainfuck "makes code hard again" in a coding culture increasingly invested in recruiting amateur developers and novice learners from more diverse and nontraditional backgrounds (210).

What's at stake for Easter is how brainfuck and other programming languages both constitute and reflect a particular kind of embodiment and power dynamic for writers and readers of code-texts. "By obfuscating human authors, such machinic performances [as those encouraged by and in brainfuck] sustain unequal access to technology and shore up the exclusive power of coding literacy," she argues (212). Easter's methodology is representative of the emergent fields of critical and rhetorical code studies, as well as broader hermeneutical, humanistic, and textual approaches to understanding how code works as a symbolic and material phenomenon. Analyses such as Easter's and compositions with code that seek similarly critical and capacious ends offer an opportunity—as well as a provocation—for teacher-scholars in English studies and other humanities fields to reimagine their learning goals, curricula, and pedagogies in light of an

increasingly ubiquitous, powerful text-based medium. While coding literacy in brainfuck is, in Easter's thinking, based on hegemonic mastery and exclusionary obscurity/illegibility, the literacy efforts for which she calls aim for "uncovering, not obfuscating." I see a profoundly pedagogical project underlying her language here that, along with the work of other scholars, informs my thinking, methods, and overall approach in this dissertation (213).[3]

Indicative of code's generative possibilities for humanistic discourse and education, the commentary on brainfuck is hardly one-sided. While Easter locates a hegemonic impulse behind the difficulty and disembodiment of brainfuck, Jacob Gaboury includes it in his discussion of the esolang phenomen in terms of a "queer imperative" and "queer computation" that "identif[ies] the ideological assumptions that produce protocologial norms and then subvert them—to make visible through a queer critical practice the values that structure our technology" (488). In Gabourey's thinking, brainfuck disrupts and critiques "our [normative] desire for a language that prioritizes clarity and functionality" (489). Irina Lyubchenko offers an entirely different take on this disruptive aspect of brainfuck, characterizing it as a computational equivalent to Giambattista Vico's notion of the poetic *ricorso*, a return to primitive ideas and forms of art that "striv[e] for direct communication [and an] awakening of corporeal imagination" as a way of resisting modern "utilitarianism and scientific materialism" as well as "skepticism that ignores subjective experiences [and] trust[s] calculative reason alone" (550). Easter, Gabourey, and Lyubchenko are part of a larger movement in humanistic scholarship that seeks to reckon with

[3] While my 'seeing' framework remains a deliberately ocularcentric conceit, Easter includes the aural as she also urges us to continue the work of making-known or making-explicit regarding "the silences sustained by machines," i.e., the "embodied difference and power asymmetries"—and perhaps here the aural better points us to the notion of narratives and stories of identities and lived experiences—that are flattened, hidden, or ignored among the alleged objectivity, neutrality, and disembodiment of the machine (213).

code in ways that both clarify and complicate its meanings while assessing its impact on lived experience. This reckoning, I will argue, is a centerpiece and legacy of what many have come to call 'the digital humanities moment' when the visibility of digital humanities work met with a broader sense of exigency around digital literacies, practices, and infrastructures.

As a field (of sorts) still deeply invested in self-definition and making space for itself among institutional and disciplinary ecologies, the digital humanities may indulge in "[u]topian visions" that energize debate over "the nature and character of humanities scholarship" while remaining vague on what its promised changes might look like as "as a blueprint" for actionable, critical *praxis* (Prescott 464).[4] Among the efforts that attempt to negotiate between utopianism and the pragmatism of blueprints, this dissertation builds a theoretical framework for teaching with code in humanities-oriented learning environments and reviews how code has functioned as a signifier in digital humanities discourses over the past twenty years. From there, I offer two case studies as blueprints—but also provocations and invitations to imagine otherwise—for how code can both bolster and transform courses that focus on writing and texts. For the collection of subdisciplines under the aegis of English departments in institutions of higher education, code provides an opportunity to advance our signature pedagogies, methodologies, and scholarship of our disciplines with respect to a novel, powerful, and increasingly ubiquitous form of technology and communication. All we need to do is see it for what it is.

[4] Prescott invokes the "utopian" descriptor not as a pejorative—as it so often is—but as a citation from the earnest language of the *Digital Humanities Manifesto 2.0* (www.humanitiesblast.com/manifesto/Manifesto_V2.pdf). While Prescott's chapter treats digital humanities utopianism as a research-based phenomenon (cf. Hirsch's findings re: bracketing pedagogy in digital humanities writing), the prominent texts on teaching in the digital humanities may very well fall victim to the same temptations. One such example is Jesse Stommel's frequently cited "Critical Digital Pedagogy: A Definition," with its trappings of Freirean idealism and claim that critical digital pedagogy is "not ashamed of its rallying cry or its soapbox...not afraid to incite, to post its manifestos, to light its torches"—in other words, that critical digital pedagogy is "as much a political approach as it is an educative one."

<Chapter Summaries/>

Chapter 1, "Toward a Vernacular Theory of Code for Digital Humanities Pedagogy," develops a deliberately *vernacular* theory of code by calling on a variety of phenomena and disciplines from popular culture to academic culture, from computer science to conceptual art. With this capacious scope I survey how code occupies our cultural imagination as a symbolic and technological phenomenon. Particularly relevant for higher education, I examine how code resonates with learning goals for the humanities, particularly in rhetoric, writing, and textual studies. I propose that code is a procedural and representational medium similar to writing, and that it leads us to reflect on the highly rhetorical decision-making processes that it enacts. To the larger question of the value of code, pedagogical or otherwise, for the humanities disciplines, I argue that the value of code lies in "seeing" it. This notion of seeing does not rest on passivity, however; I aim to establish an active and critical framework that treats code as an imperative to question, analyze, propose, provoke, and imagine in ways that exceed technical mastery of a system. Rather, I explore how seeing the code draws from critical and pedagogical traditions that view the purpose of education as the enactment of transformative justice via the unveiling and dismantling of the tools of oppression. Following this trajectory, a critical perspective on code questions the vocational instrumentalism underlying popular discussions of its socioeconomic value and its raison d'être as an educational focus. Put simply, seeing the code is a heuristic and imaginative encounter. It leads us to reflect on how a code-text was designed, how it could be designed otherwise, and how it both reflects and makes assumptions and arguments that resonate with social, political, and economic projects.

Chapter 2, "Coding (and) the 'Digital Humanities Moment,' " traces the debate on coding as it emerged from an academy still reeling from the 2008 Great Recession, when a previously niche digital humanities suddenly found itself launched into the academic mainstream. Matthew K. Gold anoints the early 2010s as "the digital humanities moment" in the inaugural edition of *Debates in the Digital Humanities* and now, roughly a decade following this pronouncement, kairotic appeals continue to color the discourse on digital humanities and coding in particular. In this chapter I begin by unpacking the implications of a provocation during the 2011 Modern Language Association national conference: that to 'count' as a digital humanist, one must know how to code. The reactions and counterarguments drew in part from familiar concerns about the purpose and relevance of humanities scholarship in an academic milieu increasingly infected with an existential sense of precarity. Beyond these echoes of a well-established overture, the debate also raised new questions about what code can and should be doing for humanities scholarship, how code allows us to think about issues both perennial and unprecedented, and what the evolving nature of code signals for the future of the digital humanities community.

Chapter 3, "Composing with Markup Language in the First-Year Writing Class," takes from the previous chapter that the mandate of the digital humanities moment is not necessarily to know how to code in a particular way, but to reckon with code as an ambient medium and emergent mass literacy. In this chapter I look to composition and writing studies as an adjacent field to critical and rhetorical code studies, taking as my cue the ubiquitous comparisons between coding and writing that find their most concentrated expression in the literacy work of Annette Vee. Discussing his approach to digital humanities pedagogy, Matthew Kirschenbaum has described code as a medium characterized by "choices and constraints," and these terms surface

frequently in the discourse about writing itself, especially when it comes to articulating the intellectual work of writing that transfers as a universal skillset ("Hello Worlds"). This notion of reflecting on one's choices and framing coding and writing as expressions of conceptual and rhetorical models provided the inspiration for the experimental approach to first-year writing that this chapter details as a case study. Overall, this case study demonstrates the value of code for humanities curricula and pedagogy and shows how "seeing the code" supports and transforms disciplinary work in composition and writing studies.

Chapter 4, "Text Encoding, Open Pedagogy, and Public Digital Humanities," looks to the more established practice of using markup language to transcribe and edit extant texts to create digital archives, collections, published editions of archival material. The overall goal of the chapter is to extend the inquiry of the dissertation—*what is the value of code, and what are its uses and impacts for humanities education?*—into a different pedagogical context within English studies that, nevertheless, remains focused on the production and reception of texts, i.e., archival work, critical and documentary editing, and digital publishing. The notion of visibility behind seeing the code turns outward in this instance as instructor-student teams begin within the physical confines of archives and special collections and complete their work on the open-access world wide web. Drawing from precedents in rhetoric and composition, library collaborations, and digital humanities, I propose an open pedagogical approach for the creation of public-facing, publicly engaged digital resources. As a model for operationalizing this approach I review a course that scaffolds student learning from the curation of archival materials to the publication of an open-access, digital documentary edition of historical correspondence.

Chapter 1: Toward a Vernacular Theory of Code for Digital Humanities Pedagogy

By exposing how the technical and the humanistic converge in the objects and processes of our everyday lives, a both/and pedagogy of code might allow our global problems to be better understood or entirely resolved.

Lauren F. Klein, "Code"

[S}cholars must engage the vernacular digital forms that make us nervous, authoring *in them in order to better understand them and to recreate in technological spaces the possibility of doing the work that moves us.*

Tara McPherson, "Why are the Digital Humanities so White?"

<Introduction/>

In a pedagogical context, theory informs our choices as teachers and influences how students encounter concepts, methods, information, and other aspects of our curricula. As part of my project's larger question—*what is the value of code for humanities education?*—this chapter begins by exploring the salient aspects of code that allow instructors to engage it purposefully and approachably for groups of learners with diverse academic and career goals. In short, this chapter seeks a vernacular theory of code for humanities pedagogy, digital or not, with a goal to equip humanities teacher-scholars to talk about, teach with, and 'do' code in pursuit of learning outcomes across the humanities curriculum. As a secondary goal, I establish a theoretical basis for the course designs and teaching scenarios that are examined as case studies in later chapters.

To develop a deliberately *vernacular* theory of code I call on a variety of evidence and disciplines: from popular culture to academic culture, from computer science to conceptual art. With this capacious scope I survey how code occupies our cultural imagination as a symbolic and technological phenomenon. Particularly relevant for higher education, I examine how code resonates with learning goals for the humanities, especially (though not limited to) rhetoric, writing/composition, and textual studies. I propose that code is a procedural and representational medium similar to writing, and that it leads us to reflect on the highly rhetorical decision-making processes that it enacts. To the larger question of the value of code, pedagogical or otherwise, for the humanities disciplines, I argue that the value of code lies in "seeing" it. Though the notion of seeing something may initially convey a sense of passivity, I aim to establish an active and critical framework that treats code as an imperative to question, analyze, propose, provoke, and imagine in ways that exceed mastery of a technical and representational system. Rather, I explore how seeing the code draws from critical and pedagogical traditions that view the purpose of education as the enactment of transformative justice via the unveiling and dismantling of the ideologies and tools of oppression. Following this trajectory, a critical perspective on code questions the vocational instrumentalism underlying popular discussions of its socioeconomic value and its raison d'être as an educational focus in colleges and universities at large. Put simply, seeing the code is a heuristic and imaginative encounter. It leads us to reflect on how a code-text was designed, how it could be designed otherwise, and how it both reflects and makes arguments that resonate with social, cultural, political, and economic projects.

Code involves many approaches that seek to address a wide variety of needs. The history of code shows a volatile ecology of languages and platforms, some of which endure despite age,

some of which flare up and burn out quickly.[5] This dissertation focuses on markup languages for their capacity to represent, transform, and create texts. In particular this capacity aligns markup languages with the learning goals and disciplinary work of composition, rhetoric, textual studies, scholarly editing, and archival studies. The later pedagogical case studies in chapters three and four dwell specifically on extensible markup language, or XML, for applications in writing studies and composition as well as textual editing and digital humanities. In this chapter I explore a theory of 'seeing the code' with vignettes from XML-driven scholarly projects as well as other examples of code and coding languages. In addition, I connect the theoretical inquiry of this chapter with critical and social justice pedagogies, especially as they are articulated in the fields of digital humanities, digital rhetoric, and computers and writing. Informed by this approach, code-based teaching and learning experiences in the humanities do not seek merely to make better users and producers with technologies (though that certainly is an effect of the process), but rather to lead learners towards more critical, imaginative encounters with digital technologies and platforms. Instead of affirming and offering entry into the powers and privileges of technocapitalism, seeing the code asks students to consider how they might transform the world for the better.[6]

As digital mediation becomes more integral to daily lives, and if code does represent an emergent mass literacy equal to reading and writing, this chapter takes seriously the need to theorize (with) code beyond the limits of technical skills and instrumental logics, and in ways

[5] For a concise history of programming geared towards an illustration of its developmental arc from physical, mechanical, and mathematical operations to abstract, symbolic, and textual composition, see Annette Vee's *Coding Literacy: How Computer Programming is Changing Writing*, MIT Press, 2017, pp. 106-113.

[6] Even the notions of betterment and transformation themselves must be decoupled from technological determinism and a late-capitalist ethos. For example, I will later interrogate the assumed virtues of speed, efficiency, and ease of use in technological developments.

that lend themselves to pedagogical praxis across the humanistic disciplines. While my thinking is inflected by my intellectual home in a department of English and my work as an administrator of a teaching and learning center, I have framed the following theoretical exploration to benefit any humanities-area or humanities-adjacent instructor who is interested in considering how code might offer opportunities to apply and even transform discipline-specific skills and methods, all the while enhancing student engagement and learning. Looking ahead to the second chapter, I conclude by suggesting that the values and purposes of coding are just as contested in the more niche area of scholarly discourse in the digital humanities. Surveying the development of those discussions during the last fifteen years, I show that the insights gained from a vernacular survey of 'seeing the code' add nuance to arguments that coding represents a means of critical building or making as well as a textual medium for critical and rhetorical analysis.

<Sight and Ciphers: Seeing the Code/>

The inspiration for a framework of seeing the code lies in cinematic and popular culture. Over twenty years after its release, *The Matrix* (1999) stands as the imaginative and cinematic preoccupation with code par excellence, even garnering a late-stage fourth entry to the cinematic franchise in 2021. Aside from the more extravagant set pieces, one understated moment in the story is particularly telling in the way that the film plays with code as a complex, contradictory phenomenon. After being awakened and unplugged from the virtual prison of the Matrix, the protagonist Neo quietly converses with Cypher, a code jockey who soon will betray his fellow hacker-liberators in the film's dystopian, machine-dominated world. If we are familiar with the

plot, Cypher's handle suggests the illegibility of his intentions as a saboteur in waiting: a foreshadowing that draws from the audience's understanding that code *obscures* meaning for outsiders and, as such, tends to provoke feelings of unease.

Gazing at an array of monitors that display the now-iconic vertical streams of green code characters, Cypher absent-mindedly muses:

> there's way too much information to decode the Matrix. You get used to it, though. Your brain does the translating. I don't even see the code. All I see is blonde, brunette, redhead. Hey, uh, you want a drink?

The scene grants the audience a respite from the frenetic pace of twists and revelations driving the first act of the film. Cypher's casual misogyny, sexualization of digital objects, and booze-enabled gesture of masculine bonding anticipate many of the cultural problems of the IT industry that now draw widespread scrutiny.[7] More importantly, though, Cypher suggests a hermeneutics of code akin to the work of the Platonic rhapsode: the speaker whose skillful delivery comes not from technical mastery or experience but rather from divine inspiration: the speaker as a vessel through which the philosophers and poets speak.[8] Apprehending the meaning of code without actually seeing it, grasping the signified without interpreting—or decoding—the signifier, Cypher figuratively extends the brain-body dichotomy that the film's premise literalizes.

---

[7] A 2017 Department of Labor investigation into Google, for example, "found systemic compensation disparities against women pretty much across the entire workforce," and in the midst of this investigation a disgruntled male engineer released a manifesto that quickly went viral within and beyond the company, alleging Google's "moral bias" of leftist politics and insisting on essential "biological" differences between men and women that determine their place and role in the IT workforce (Levin; Conger). Anastasia Salter and Bridget Blodgett explore the sexism and misogyny endemic to the IT world in *Toxic Geek Masculinity in Media: Sexism, Trolling, and Identity Policing*, Palgrave Macmillan, 2017.

[8] Cf. Socrates's final speech in Plato's *Ion*: "If you're really a master of your subject, and if, as I said earlier, you're cheating me of the demonstration you promised about Homer, then you're doing me wrong. But if you're not a master of your subject, if you're possessed by a divine gift from Homer, so that you make many lovely speeches about the poet without knowing anything—as *I* said about you—then you're not doing me wrong" (34-35).

In a modern context, the scene presents Cypher in a strikingly similar light in which we now might view the archetypal " 'lone genius' coder" (Thompson). We imagine the avatars of Silicon Valley as both oracle and hierophant for the twenty-first century: disruptive entrepreneurs and rogue hackers who call forth a new vision for the future and, at the same time, command the ability to interpret the esoteric systems that so powerfully affect our day-to-day, lived experience. Cypher may not even see the code, but all we can do as *viewers* of the film, appropriately, is *see code* as an aesthetic object both mythic and futuristic: an alchemy of historically evocative and inscrutable rune-like characters that, somehow, constitute an advanced technological dystopia for billions. Code enacts the insider-outsider dynamic that it served long before modern computers: an attempt to obscure legibility so that only an intended audience may be able to interpret messages. Alan Turing's famous efforts to decode the German Enigma during World War II, for example, leveraged machine algorithms to decipher encoded communications from one person to another regarding the positions and missions of military assets. Along more innocent lines, childhood friends may develop their own secret language that no one else may understand as an affirmation of deeply felt kinship. On the one hand, then, code looms in the popular imagination with a raison d'être to conceal: for its meaning to *escape* understanding, except for the (s)elect few.

Yet, as Cypher also suggests, computer code can be learned, applied, and translated. As a language, code must follow rules: grammar, syntax, nomenclature, and so on. Some scholars have gone as far as suggesting that learning a programming language (e.g., Python, JavaScript, .NET, PHP) should fulfill the language requirement that persists, often in a vestigial capacity, in the curricula of many United States humanities graduate degree programs (Kirschenbaum "Hello

Worlds"). The idea of code and programming as the emergent language of our time reverberates across major texts and authors in digital studies; Lev Manovich, for instance, has sweepingly opined that "[s]oftware has become a universal language, the interface to our imagination and the world." It's important to note that the enthusiasm for software as a universal language elides that programming emerged from and remains globally dependent on the English language. In chapter two I historicize the development of computer programming as an English-based profession during the 1950s and 60s, and I also explore coding experiments that use non-English languages as a basis for programming. Even in specialized discourses such as linguistics, speech pathology, and cultural anthropology, language quickly becomes a slippery concept, and even more so in popular discourse on computer programming. Analogizing computer code with human language does not usually claim an exact equivalency but rather stakes a claim about the contested value of code as a technological and cultural phenomenon. In other words, it marks code as something we humanists have a right and imperative to study. Yet, some critics nevertheless resist the comparison and warn against normalizing programming among the more traditional humanistic pursuits. This critique usually urges a careful evaluation of the rhetoric ostensibly bolstering the legitimacy of the digital humanities against implicit allegations that the mainstay of humanities work has become disconnected from the zeitgeist and suffers from outdated perspectives and methodologies. According to this reasoning, we risk amplifying the pressures already mounting against the humanities and accelerating the erosion of funding for research, hiring, programs,

teaching, and student support in humanistic fields, programs, or projects that do not foreground the digital as the focus and/or means of study.[9]

Nuanced arguments highlight the opportunities and risks following the ways in which we position the digital with respect to the longstanding goals of humanities scholarship. In the inaugural volume of *Debates in the Digital Humanities* (2012), Matthew Kirschenbaum describes *digital humanities* as "a tactical term" that is "unabashedly deployed to get things done" in institutional ecologies ("Digital Humanities As/Is a Teactical Term," emphasis added). Drawing from Kirschenbaum's proposition, Jim Ridolfo and William Hart-Davidson frame their 2015 collection *Rhetoric and the Digital Humanities* as a way of showcasing how the digital humanities can be used to advance the agendas of rhetoric studies, technical and professional writing, technical communication, and computers and writing, the latter of which has investigated digital media and communication practices since the late 1970s and early 1980s (3-4). In another line of thinking, the ubiquity and tone of manifesto-like definitions in digital humanities discourses risks othering the ostensibly non-digital humanities as out-of-date and out-of-touch: an allegation that may even align with the political right and the neoliberalization of higher education (Golumbia, "Death of a Discipline" 170-71). The concern is not whether the humanities will lose a sense of relevance in the abstract, but the degree to which political, economic, and organizational infrastructures may shift—perhaps permanently—towards or away

[9] In the inaugural volume of *Debates in the Digital Humanities*, Matthew Kirschenbaum describes digital humanities as "a tactical term" that is "unabashedly deployed to get things done" in institutional ecologies ("Digital Humanities As/Is a Teactical Term"). Drawing from Kirschenbaum's proposition, Jim Ridolfo and William Hart-Davidson frame their 2015 collection *Rhetoric and the Digital Humanities* as a way of showcasing how the digital humanities can be used to advance the agendas of rhetoric studies, technical and professional writing, technical communication, and computers and writing, the latter of which has investigated digital media and communication practices since the late 1970s and early 1980s (3-4).

from areas of humanistic and liberal arts education that do not foreground technology in ways that resonate with policymakers and other stakeholders with infrastructural and political agency. In a 2019 *New York Times* op-ed, for instance, the director of the Alexander Grass Humanities Institute at Johns Hopkins warns that the increasingly common paeans to coding may result in tangible, lasting impacts on humanities education at the level of policy, especially in light of proposed bills at state and federal levels which would allow learning a programming language to fulfill the foreign language requirement for high school graduation (Egginton). Beyond the niche group of digital humanities scholars who see a lost opportunity to have better designed their educational journeys (i.e., if a coding language had fulfilled their language requirement), those who desire to enter the business and IT sectors may also find in this proposition an opportunity to focus on career-ready skills and declutter their curricula with respect to the shibboleths of a liberal arts education. Egginton warns against a false equivalency with computer languages that downplays how the study of natural languages —and, thus, the cultures and ideas that produce them and are produced by them—facilitates "learning who we are by interacting with others."

Indeed, code itself is not written *to* another person, at least not in the same way as are the forms of communication that we traditionally study in the humanities. Rather, code instructs an assemblage of software and hardware to act in particular ways when conditional inputs and requirements are satisfied. Crucially, David Golumbia explains, computer languages thrive on "univocal, correct, 'activating' interpretations" and simply cannot function with the "ambiguity, context, and polysemy" that are a fundamental condition of human language (*Cultural Logic* 84). At first glance, this process seems not to lend itself to the cultural inquiry and self-discovery to which Egginton and other defenders of the humanities allude. Yet, as many digital humanities

scholars assert, code nonetheless exerts agency on both culture and individuals to the degree that we can speak not only of its general value as an object of interest for humanities curricula, but also of its rhetoricity and aesthetics as a system of meaning-making. Kevin Brock's 2019 monograph *Rhetorical Code Studies* pursues this thesis programmatically. Drawing from Estee Beck's theory of persuasive algorithms and positioning his inquiry among digital rhetoric, computers and writing, software studies, technical communication, and critical code studies, Brock places code in an expansive rhetorical ecology of developers, users, and systems from the material to the cultural. Studying code in this way, Brock argues, leads us to "understand the creative processes we engage in regularly as part of our humanistic activity" (182). Not exactly a 'natural' language in the traditional sense, code nevertheless prompts us to reflect on who we are, how we understand the world, and how we engage with others.

<The Rhetoricity and Aesthetics of Code/>

In the critical introduction to the experimental poetry collection *Moonbit*, which uses the Apollo Space Program's guidance computer code as material for selection and transformation, James Dobson and Rena Mosteirin explore "the affordances and limitations of a language that is machine-oriented yet human-authored" (15). Code might be written to a computer, but it also is written by people, with others in mind, as part of a program or algorithm that exerts agency onto others. Code even further can make metatextual arguments about the medium itself, especially in the case of more experimental, creative, or "weird" languages, which "make it undeniable that programming is a way to express things to humans, not only to machines" (Easter, "Fully

Human" 203). Put another way, code instructs a machine to perform actions, but we also know that others will read, borrow from, form ideas, and take action with our code. According to Dobson and Mosteirin, code "is not written just for a computer…[it] has many audiences and can be shaped into several different forms" (18). Their understanding of codework is informed significantly by Caroline Levine's new formalism, in particular how Levine proceeds from a more capacious understanding of form as "an arrangement of elements—an ordering, patterning, or shaping,"[10] which allows her to speak of the aesthetic and social as well as the political and economic as constitutive of formal phenomena. Levine does not address code, but her language is striking for its resonance with codework. A type of code, markup language, is quite literally an "arrangement of elements" with containers that follow an ordered hierarchy, and the notion of shaping, patterning, and arranging evokes digital humanists' notion of code as modeling (3). In other words, the formal elements of code—how an algorithm solves a problem, how data is input and transformed, how a process is executed, how the output is represented and manipulated by an end user—implicate social, material, economic, and political forms as well.

What new formalist and rhetorical accounts of code share, then, is the sense that we write code knowing that what it instructs machines to do may have profound effects on people who may or may not be aware of how code affects their lives. While *Moonbit*'s use of the Apollo

[10] Dobson and Mosteirin's use of "codework" as opposed to "code" draws from John Cayley's terminology, which looks beyond the code-object itself to emphasize "the emergence of new or less familiar rhetorical strategies" that account for expanded notions of audience and text in codework, as well as code's inextricable reliance on "strict logical process" and "compilation in the programmer's sense," all of which implicate "unacknowledged tropes and figures" of art—i.e., *poiesis* or making—in digital media. In the same issue of *Electronic Book Review* in which Cayley's essay appears, Rita Rayley defines codework as "the use of the contemporary idiolect of the computer and computing processes in digital media experimental writing" with a goal "to bring the function and code of the computer to a kind of visibility." Earlier theories of the aesthetics and rhetoric of code often draw from experimental media and poetry, with an emphasis on code as a "generative" performance that unfolds somewhat autonomously from its authoring (Cox, McLean, & Ward 162).

Guidance Computer (AGC) language offers perhaps the least ambiguous instance of trusting human life and wellbeing to the workings of code, a more recent and earthbound case involves Amazon abandoning its attempt to automate job applicant rankings in 2018 because the program was designed to judge applicants based on data from past hiring decisions, thus perpetuating any biases, particularly against women, that informed them (Dastin). Similarly, researchers found that a widely used criminal risk assessment software underpredicted white defendants' likelihood of recidivism while overpredicting the likelihood for Black defendants—*even more so* than an experimental group of non-experts responding to an online survey (Dressel and Farid). A study of three commercial gender classifiers using facial analysis algorithms found that all three were notably more accurate for male and lighter-skinned faces (Buolamwini and Gebru 8). Chatbots built on neural networks using sophisticated natural language processing have engaged in hate speech or, conversely, refused any discussion involving race (Schlesinger, O'Hara, & Taylor 2).[11] In these cases, the mythic objectivity of math and machines is revealed to operate on the value propositions and interpretations that are the stuff of rhetoric and ideology.

Digital humanities researchers accordingly take great pains to consider the determinants of coding projects while writing code for others to read, reuse, and repurpose. For archival work,

[11] In his monograph *Critical Code Studies*, Mark C. Marino reflects that he set down the path to define such an area of study as he sought a framework for analyzing chatbots in a way that foregrounded their unique properties as digital-textual phenomena (18). For a primer on the racial bias of natural language processing algorithms, see Eleanor Shearer, et. al., *Racial Bias in Natural Language Processing*, Oxford Insights, 2019, www.oxfordinsights.com/racial-bias-in-natural-language- processing. The subject gained national attention in early December 2020 when Timnit Gebru, co-lead of Google's AI ethics team, was fired just as the company rejected use of her name and affiliation on a forthcoming paper on the problems of larger datasets in natural language processing (the engine of most of Google's efforts, including its prized search algorithms). The paper's critique concerns not only that training AI on broad swaths of language from the internet will "obviously" include undesirable uses and ideologies, e.g., racism, but also that "nuances" of emergent rhetoric and vocabulary will be lost (such as the "shifts in language" led by the MeToo and Black Lives Matter movements) and that trained AI will exhibit bias towards the largest "linguistic footprint[s] online...reflecting the practices of the richest countries and communities" (Hao).

researchers follow standards developed by the Text Encoding Initiative (TEI), a consortium of digital humanities and textual scholars that publishes guidelines for encoding documents with extensible markup language (XML) so that other researchers may borrow from, adapt, transform, or otherwise manipulate a code file. Later chapters explore the TEI and XML in more depth, but the essence of this approach to coding lies in elements or tags (included among and containing the transcribed text of a document, as well as in freestanding 'header' sections) that follow standardized metadata formats allowing them to be read, mined, manipulated, and even revised by researchers beyond the immediate team or project for which a coding scheme was applied.

```
<element>This is a basic example of an element containing text.</element>
```

In simpler language, TEI projects follow standards because researchers *want* others to access, understand, and transform their code: a visibility that also suggests malleability-by-design.[12] To see the code is also to be invited to use it and make it one's own.

As a venue for meaning-making, this particular rhetorical triangle suggests similar premises and unfurling complexities to rhetorical models for teaching and analyzing writing in so-called 'natural' language: a distinction at once misleading *and* pleonastic were it not the apparent opposite of machine language, i.e., code.[13] As Estee Beck explains of this unique rhetorical situation, "the context of the production of code ruptures from an authorial moment of

[12] Similar to malleability, researchers have emphasized the importance of *tuning* or tunability as a necessary step beyond mere transparency in code: a "probing and adjustment" that seeks to "refine" the output of neural networks that power applications such as AI chatbots (Schlesigner, O'Hara, and Taylor 9).

[13] The apparent dichotomy of natural and machine language, especially with respect to the contradictory senses of "natural" as both inaccurate *and* so obvious as to appear redundant, emerges from the limitations of Enlightenment humanist frameworks that privilege particular cultural—and cultures'—activities and artifacts as indicative of an idealized "human nature." How digital humanists can avoid this trap without abdicating their work to social science or data science is a concern that undergirds this dissertation, hinging on signature epistemologies, methodologies, and pedagogies of the humanities as they might be identified independently from particular objects of study.

creation" in the sense that each iteration of the code represents a "residual performative act within precise [i.e., changing] contexts" while both readers of the code and users of the interface (that the code produces) enact another layer of "contexts or iterations" that continue to proliferate. With this framework Beck is able to write about the "performativity," and, thus, the rhetoricity of code, which does not convey meaning as a static signifier as much as it enacts processes that produce meaning anew in differing and sometimes unexpected ways. A purely machine-oriented sense of code, by contrast, would emphasize replicability, iterability, and unchangeable meaning in the context of recursive and rapid calculations. Here we arrive at perhaps one of the foundational theses of code for the humanities teacher-scholar: code is written for computers and people, and may exert a significant—and evolving—influence on the latter.

The metatextual example of the TEI merits further unpacking given that the organization seeks to theorize and model text itself as well as the documents and communicative acts that text constitutes. Beyond the instrumental and rhetorical value of extensibility (i.e., allowing for other scholars to do new things with an XML text/file through a standardized markup syntax and nomenclature), the TEI seeks to draw from and refine theories of the document, the text, and scholarly editing in order to represent textual objects and other material artifacts with rigor while providing flexibility for the needs of specific projects. In this instance the study of code entwines especially with the study of language and writing. For the development as well as the application of TEI guidelines, we ask how systems of representation model what they purport to represent, as well as how people, events, and other forces have shaped those systems, and to what ends. We ask about the effects of these systems on the agents who use them, the networks in which they are used, and, broadly, the possibilities for understanding and expression that they both enable

and constrain. To put it another way, how do these systems of representation, as well as our study of them, help us to understand the human (and perhaps even posthuman) condition?

If we place code in the purview of language, literature, and writing studies, functionality and mechanics join aesthetics, form, ethics, and philosophy in a way that allows us to speak of code, inscrutable and instrumental as it may seem, as if it aspires towards elegance and beauty. These tropes of craft and artistry, in fact, have become commonplaces for programmers and in discourses on programming. They form the basis of Andy Oram and Greg Wilson's *Beautiful Code*, a collection of essays written by programmers reflecting on "the most beautiful piece of code" they have encountered. In the preface, Wilson relates his own moment of awakening when he realized that code "could be as elegant as well-made kitchen cabinets, as graceful as a suspension bridge, or as eloquent as one of George Orwell's essays" (xv). Yukihiro Matsumoto, a creator of the Ruby programming language, writes in his chapter not incidentally titled "Treating Code As an Essay" that the design of Ruby aspires towards beauty through brevity, familiarity, simplicity, flexibility, and balance (478-81). Figure 1 below shows Matsumoto's comparison of the "Hello World" program in Java and Ruby.

Java	Ruby

class Sample {}
 	
public static void main(String[] argv) {
System.out.println("Hello World");
}
	print "Hello World\n"

Figure 1: Matusmoto's comparison of "Hello World" in Java and Ruby to illustrate the aesthetics of brevity (478)

While programmers such as Matsumoto take great pride in their creations, others may feel disappointment if they perceive their work to be a failure of craft. Researchers, for example, may

withhold their code out of "embarrass[ment]" that it is "ugly," perhaps designed without much of a public audience in mind and written by a researcher with a functional coding literacy but who would not consider themselves a skilled coder. This sort of self-censorship may present a failure of scientific ethics given that a project's computational methods become something of a black box when the code is hidden away as if it were a mess swept under the rug. It also may present a failure of sustainability due to the likelihood that the code will be lost, rendering a researcher's methods irreplicable without a full demonstration of how their analyses were performed (Hsu). This concern over appearances represents an irrevocable condition of writing or using code for *any* project beyond computational research; as code becomes an object for any kind of public readership, composing it always involves an anticipation of how it will be received and reused.

Beautiful or ugly, code aesthetics are a perennial topic of debate, concern, and in some cases, ridicule. In the popular webcomic series *XKCD*, Randall Monroe riffs on programmers' aspirations towards elegance (and spectacular failures to realize it) in "Code Quality" and "Code Quality 2." In the first, a skilled programmer critiques the work of a self-taught amateur with increasingly hyperbolic assessments. "It's like someone took a transcript of a couple arguing at Ikea and made random edits until it compiled without errors," she concludes, as the berated novice skulks off to consult a style guide. The sequel continues the insult comedy with more inventive barbs, after which our berated amateur retorts that his code "runs fine for now." Always getting in the last word, our expert offers a final rejoinder: "So does a burning bus." We see an excerpt of "Code Quality 2" in Figure 2 below.

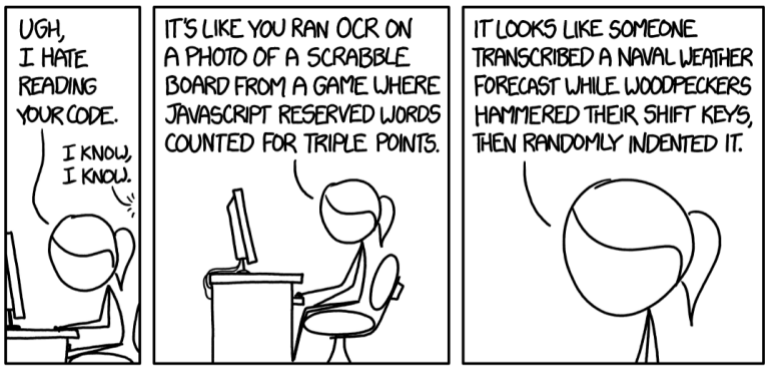


Figure 2: panel excerpt from Randall Munroe's "Code Quality 2," xkcd.com/1695/

The comics remind us that while aesthetics and functionality are related, they are not always the same. Munroe's insult comedy rests on the premise that code occupies an aesthetic field, and that code, like language, might "run fine" in the sense that it results in the desired output or goal, but the means to attain its end may not represent the most intuitive, artful, and efficient approach. Writers and teachers of writing seek a similar understanding: beyond the conveyance of information, how does the communicative act engage the most successfully and reflectively in a given rhetorical situation?[14] The burning-bus punchline evokes an imbalance in which the code's functionality is entirely unsustainable; in other words, it will work only for a very short time. Or, it might work, but no one would want to use it. For writers and scholars of writing studies, it may further evoke memories of particular first drafts, and the reference to a style guide resonates with the need to balance individual and collective ethos in academic and professional writing. Perhaps the most explicit invocation of the writing style guide is Brian W. Kernighan and P. J. Plauger's *Elements of Programming Style,* first published in 1974, which

[14] There are many coding languages, and more continue to crowd the field. While Paul Ford uses the "tool" analogy to explain why one might choose a particular language given that they all appear to accomplish the same goal, i.e., make the computer do what one wants, he also suggests that different languages "represent the world in different ways, structure data in different ways, and address the components of the computer in different ways." Ford's sense of code suggests ontological and epistemological dimensions, as code makes arguments concerning the defining nature of things and structures our knowledge about them.

pays homage to Strunk and White's *Elements of Style* not only in overall title but also in the maxim-format of its section headings, e.g., "Avoid too many temporary variables." From all of this, the elegance of code emerges not as a *purely* aesthetic judgment—or, better yet, it suggests a unique aesthetics of code based on measures of accuracy, efficiency, intuitiveness, flexibility, and extensibility with both human and machine audiences in mind.

<Encoding Correspondence: The TEI and the Elegance of Code/>

As a demonstration of how we perceive and operationalize the elegance of code, I'll briefly review the Text Encoding Initiative's development of a metadata structure and lexicon for encoding correspondence. According to its website, "[t]he Text Encoding Initiative (TEI) is a consortium which collectively develops and maintains a standard for the representation of texts in digital form. Its chief deliverable is a set of Guidelines which specify encoding methods for machine-readable texts, chiefly in the humanities, social sciences and linguistics." Among many types of documents that are being digitized and encoded, letters and correspondence have become an important case study for the encoding of texts and the associated publication of databases and digital scholarly editions. "Crucial for both digital editions and collections of metadata," writes Gabriel Hankins, "will be shared standards, practices, and platforms for the representation, preservation, and communication of digital forms of literary correspondence." Thus, letters present an opportunity to test, apply, and refine various approaches to theorizing textual objects and representing them in code. Letters have become an item of particular interest to test and demonstrate the affordances of digital encoding and scholarly editions because (a)

they often are held in multiple, “far flung” locations with an “equally dispersed” audience of archivists, scholars, commercial interests, and lay enthusiasts; (b) printed editions are incapable of containing the contextual and associated artifacts that defined the social networks in which the letters operated; and (c) the study of letters in codex form “essentially reduces and thus misrepresents the relational aspect of correspondence” (Hankins). Implicit in Hankins’s list is the notion of correspondence as uniquely *in situ*, historically and rhetorically, which presents the challenge of modeling a text both self-contained and always gesturing outward from itself. In other words, the nonlinearity, referentiality, and contextuality of correspondence align with the computational and representational capabilities of code and markup language in particular.

In 2013, the TEI special interest group for correspondence established a task force with the charge “to better support the encoding of correspondence” at a metadata level. Consisting of Peter Stadler, Marcel Illetschko, and Sabine Seifert, this task force first looked to examples of correspondence metadata models in noteworthy archival and editorial projects. Of these, the Digital Archive of Letters in Flanders used the following metadata structure for encoding its correspondence.[15]

```
<dalf:letDesc xmlns:dalf="http://ctb.kantl.be/DALF/2.0">
  <dalf:letHeading xml:id="KVDW.part" default="true">
    <dalf:letAuthor ref="names.xml#KVDW" attested="yes">Karel van de Woestijne
      </dalf:letAuthor>
    <dalf:letAddressee ref="names.xml#EDB" attested="yes">Emmanuel de Bom</dalf:letAddressee>
    <dalf:letPlace attested="no"/>
    <dalf:letDate notAfter="1904-07-25" notBefore="1904-08-02">na 1904-07-25 en voor 1904-08-
      02</dalf:letDate>
  </dalf:letHeading>
  <dalf:envOcc occ="false"/>
  <dalf:figOcc quantity="1"/>
```

[15] ctb.kantl.be/project/dalf/

```
  <dalf:type>Brief, geen omslag</dalf:type>
</dalf:letDesc>
```

Before analyzing the excerpt, a few notes on the syntax and nomenclature of XML are merited. *Elements* are the fundamental unit of markup language and enclose strings of text—either about or from the document—that they describe. Designated with open and close brackets, elements may contain any number of other elements that themselves may contain any number of other elements. The nested structure of elements is known as the ordered hierarchy of content objects (OHCO) by which XML models the structure of a text. In the examples here (as well as in most coding applications) indentation clarifies the hierarchy of container-and-contained relationships between (blue) elements, and an element 'closes' when its tag is repeated with the addition of a backslash. A family-tree lexicon describes the relationship between elements (and the data or features that they describe) in this hierarchy. *Sibling* elements share the same container element but do not contain each other; `<dalf:letPlace>` and `<dalf:letDate>` are examples of siblings in the above code excerpt. Both of these share the *parent* element of `<dalf:letHeading>` and the *ancestor* element of `<dalf:letDesc>`. Put another way, `<dalf:letHeading>` is the *child* of `<dalf:letDesc>`, while `<dalf:letPlace>` and `<dalf:letDate>` would be its *descendents*.

Nested in the parent element `<dalf:letDesc>` or the letter description, `<dalf:letHeading>` or letter heading contains child elements for the author, addressee, place of composition, and date of composition. Sibling elements to `<dalf:letHeading>` establish the occurrence of envelopes and illustrations, as well as a classification for the letter. Within the elements, *attributes* (in orange) establish *values* (in brown) for unique identifying information, references to other metadata elements (in this case, person entries), dates, editorial verifications, occurrences, and quantities. The (blue) element structure encloses the metadata text itself (in

black), otherwise known as parsed character data. Using this example, as well as the Carl Maria von Weber: Collected Works project,[16] the special interest group developed the `<correspDesc>` or correspondence description element along with an accompanying structure and nomenclature for encoding correspondence metadata in TEI projects. (Again, the tripartite structure of XML is element, attribute, and attribute value.)

```
<correspDesc>
  <correspAction type="sent">
    <persName>Carl Maria von Weber</persName>
    <settlement>Dresden</settlement>
    <date when="1817-06-23">23 June 1817</date>
  </correspAction>
  <correspAction type="received">
    <persName>Caroline Brandt</persName>
    <settlement>Prague</settlement>
  </correspAction>
  <correspContext>
    <ref type="prev" target="http://www.weber-gesamtausgabe.de/A041209">Previous letter of
      <persName>Carl Maria von Weber</persName> to <persName>Caroline Brandt</persName>: <date
      from="1817-06-19" to="1817-06-20">June 19/20, 1817</date>
    </ref>
    <ref type="next" target="http://www.weber-gesamtausgabe.de/A041217">Next letter of
      <persName>Carl Maria von Weber</persName> to <persName>Caroline Brandt</persName>: <date
      when="1817-06-27">June 27, 1817</date>
    </ref>
  </correspContext>
</correspDesc>
```

The precedents for `<correspDesc>` were chosen for their extensive documentation and the close degree to which they cleaved to the established TEI guidelines while also customizing them to fit the needs of the project at hand. Stadler, Illetschko, and Seifert's revised model demonstrates how the criteria for elegance in code—accuracy, efficiency, intuitiveness, flexibility, and extensibility—can be applied to markup language.

[16] www.weber-gesamtausgabe.de/en/Index

Accuracy and intuitiveness in this case lead us to the theoretical task of modeling texts and forms. In other words, to what degree do both metadata structures represent an accurate and intuitive theory of the text and, in this case, the particular form of correspondence? The lexical choices of the root elements establish divergent theories for their content; `<dalf:letDesc>` points to the *letter* as document while `<correspDesc>` suggests *correspondence* as a communicative act. For the former, a letter heading element contains children that ask the typical questions for a letter: who wrote it, and to whom did they write it? where and when was it written? Additional elements continue to model the letter in terms of its material constitution: is it enclosed in an envelope? does it contain drawings or illustrations? By contrast, `<correspDesc>` conceives of the letter as a "communicative act" in a discursive and social context. The "heart" or "atomic unit" of `<correspDesc>`, we are told, is the `<correspAction>` element, which categorizes bibliographic information such as people, places, and dates according to the *actions* to which they correspond. Leveraging the common attribute `@type` allows for the `<correspAction>` element to describe who, when, and where the letter was sent and received.[17] In the TEI Guidelines, suggested values for `@type` in `<correspAction>`—sent, received, transmitted, redirected, forwarded—enables even more flexibility for encoding the social and material vectors of correspondence.

In addition to theorizing correspondence as encompassing an array of actions, Stadler, Illetschko, and Seifert propose that it occurs in a "continuum" or sequence of communication. The `<correspContext>` element, sibling to `<correspAction>`, accordingly contains `<ref>` tags that

[17] In XML, each element may be further specified with any number of attributes, each of which is assigned a value. The TEI Guidelines explicitly establish the nomenclature for attributes as well as the kinds of attributes that may be used for particular elements, but the values for those attributes are either merely suggested or left entirely to the discretion of the project. In short, the three parts of XML are `<element attribute="value">`. For clarity, I will use the @ prefix throughout the dissertation to indicate when an attribute is being named, and I will enclose elements in less-than and greater-than (< >) characters. Markup will be further distinguished with a sans-serif font.

point to previous and subsequent letters that place it in the context of the larger conversation that the letters enact as a corpus. The `@type` attribute allows the `<ref>` element to specify either previous or subsequent letters, while the `@target` attribute provides the location of the referenced documents.[18] Stadler, Illetschko, and Seifert explain that `<correspContext>` is especially useful for projects that involve correspondence "where different letters written on the same day may be part of different discussions with different addressees [sic], or where one and the same (forwarded) letter generates different answers from different writers." For projects that engage with larger bodies of texts, `<correspContext>` maintains clear lines of communication with a single entry in the header (i.e., the metadata) for each document. Overall, `<correspDesc>` enacts a theory of the text defined by actions in sequence among material and discursive networks: e.g., composition, transmission, and reception for the material and information exchange, address, and reply for the discursive.

While this move from the letter-as-document to correspondence-as-act seems at first to lose the material metadata that the DALF recorded—envelope and other materials, illustrations in the letter text—the `<teiHeader>` element that contains all of a document's metadata (including the `<correspDesc>` section) already provides a `<physDesc>` element for recording the physical state of the document, an `<accMat>` element for describing and linking to any accompanying materials (such as an envelope or enclosure), and an `<msContents>` element for summarizing the textual content of the document. Within any of these, the `<figure>` element can indicate visual

[18] In this case, URLs are used as the unique identifier for the locations of the referenced documents, but other options include using the unique identifiers established within the XML itself with the `@xml:id` attribute. Should the digitized records for the documents be held in a stable, sustainable, and open-access location, URLs make a sensible choice for the value of the `@target` attribute in the `<ref>` element, while other projects may choose to cite internally if files held at URL locations do not meet the criteria of stability, sustainability, and accessibility.

representations, charts, or illustrations with an `@xml:id` value that corresponds to how it is tagged in the `<body>` of the text. By way of efficiency and flexibility, then, the `<correspDesc>` structure makes good use of the `<teiHeader>` markup and avoids redundancy—a best practice for both human and machine audiences—by adding only the unique aspects of correspondence (theorized as material and discursive act) not already in the purview of extant metadata categories. Even further, the addition of `<correspDesc>` to the TEI Guidelines minimized the number of new tags (i.e., adding only `<correspAction>` and `<correspContext>`) in a nested structure that relies on the flexibility of extant elements (e.g., `<persName>`, `<date>`) and attributes (e.g., `@type`). Whereas the DALF metadata most notably places elements in sibling relationships, `<correspDesc>` takes advantage of XML's categorical syntax (and logic) by placing elements in filial relationships (e.g., the information added to a plain `<persName>` element by placing it within `<correspAction @type="sent">`). Thus, the structure of XML itself provides an additional layer of metadata; an element's ancestors, parents, siblings, children, and descendants affect our understanding of its conceptual and relational context.

Lastly, Stadler, Illetschko, and Seifert developed `<correspDesc>` with an eye on the need for extensibility in response to "a general and growing demand from correspondence projects for interchange and linked-data capabilities." In other words, because of the proliferation of digital texts and encoding projects, the metadata itself has become an object of aggregation and mining in addition to the marked-up texts themselves. One such database, correspSearch, allows users to "search within the metadata of diverse scholarly editions of letters...according to the letter's sender, adressee [sic], as well as place and date."[19] Stadler, Illetschko, and Seifert have

[19] correspsearch.net

facilitated the interoperability of correspondence metadata with what "is essentially a constrained subset of the full TEI standard," a Correspondence Metadata Interchange format, which itself draws from standards for the interchange and interoperability of various data types such as people, places, and dates.[20] In addition to accuracy, efficiency, intuitiveness, flexibility, and extensibility, the elegance of `<correspDesc>` may alternatively be described as a contextual awareness that simultaneously considers incrementally larger purviews: from the text(s) of a project, to the code that the project uses, and finally to the network of projects featuring similar texts and/or using similar encoding systems.

As far as coding languages go, XML is of a notably high level. The altitudinal distinction of higher or lower-level programming languages marks the degree to which computing processes are abstracted, automated, or concealed, as well as the degree to which natural language and human-readable instructions are employed. In fact, the computer does not 'know' what to do with XML at all given that the markup language functions descriptively (i.e., it does not tell a computer what to do with any of the data that is tagged with elements). This separation of descriptive encoding from the code that is used to process and present that data in an electronic environment (e.g., a website via a browser) is fundamental to editorial practice, but only with the digital medium can we "fully comply" with the principle of separating and documenting editorial interventions apart from the published form in which an audience encounters the text(s) (Pichler and Bruvik 182). In the context of digital editing and scholarly editions, then, *seeing* the code is precisely the point as opposed to 'doing' something with it, whether manipulating, processing, or

[20] The Integrated Authority File (www.dnb.de/EN/Standardisierung/GND/gnd.html) and the Virtual International Authority File (viaf.org) provide standards for the interoperable encoding of people and places, while the W3C format (www.w3.org/TR/NOTE-datetime) provides standards for the interoperable encoding of dates.

translating the code. As a record of all editorial transcriptions, interventions, and frameworks for representing a body of texts, the XML corpus file itself (and not, as it may first seem, the digital publication, website, or other 'final' form) is the prized product of digital scholarly editing. This distinct aesthetic of markup language therefore enacts a hermeneutics and value system based on description, recalling Susan Sontag's famous dictum that "[i]n place of a hermeneutics we need an erotics of art" (14). Perhaps we also need an erotics of code.

<Epistemologies of Code: Choices, Constraints, and Content Objects/>

Distinguishing high and low levels of code builds from the fundamental understanding that humans and machines perceive code differently and have different purposes when it comes to doing things with code. To what degree does a machine—a complex network of databases, software, and hardware—possess a notion of elegance when it comes to what is written for it or presented to it as input? Does it cleave to or diverge from criteria that make code appear both useful and well-crafted for the people who write it? On the one hand, computers approach code in a way remarkably alien to what goes through the mind of, say, Munroe's frustrated coder from *XKCD*. Paul Ford memorably describes computers as "clock[s] with benefits...doing second-grade math, one step at a time" in a significant example of public-facing technical writing that, as its title suggests, attempts to answer the deceptively simple question: "What Is Code?" And Ford's ambition lies beyond definition and science literacy for its own sake; the clear exigency is that code *does things* to us and our environments well beyond solving a technical problem or executing a technical process. Geoff Cox and Alex McLean explain this as the "excess" of code

according to speech-act theories, "where context is lost or turns in on itself recursively." In other words, being able to write and run code (or to explain how code works from a technical standpoint) tells only part of the story; "[w]hat is important is the relation to the consequences" of the coding/speech act (38). To that point, Mark Graham, Matthew Zook, and Andrew Boulton build from a notion of "code power" to explain how our experience of *physical* urban spaces is mediated and manipulated by code in "a very centralised yet hidden manner" (470). Against the outdated distinction between virtual and physical spaces, Graham, Zook, and Boulton propose that our phenomenological realities moving in and through space are "augmented" by digital information and algorithms that emerge from and reinforce vectors of power: social, political, economic, and so on (465). While code in *The Matrix* produces experiences of virtual space, Graham, Zook, and Boulton show that code, in fact, produces our experiences of *real* space in urban geographies. Their insights reveal some of the stakes behind Ford's argument that "[e]very month, code changes the world in some interesting, wonderful, or disturbing way."

Humans and machines encounter code differently, but how does a computer perceive code, and what does it do with code? According to Ford, "[c]omputers usually 'understand' things by going character by character, bit by bit, transforming the code into other kinds of code as they go." To explain, he narrates what a computer would 'think' processing the subroutine "Hello Nerds," a modulation of the ubiquitous "Hello World" program: `PRINT{HELLO NERDS}`:[21] In Figure 3 below, Ford illustrates `PRINT{HELLO NERDS}` as a step-by-step process.

[21] Apropos of Cox and McLean, the first subroutine that a student learns how to write in computer science simulates a speech act and the exertion of agency on/in the world: the text output of "Hello World."

Character	Meaning
P	*Hmmmm...?*
R	*Someone say something?*
I	*I'm waiting...*
N	*[drums fingers]*
T	*Any time now...*
Space	*Ah, "PRINT"*
{	*String coming!*
H	*These*
E	*letters*
L	*don't*
L	*matter*
O	*la*
Space	*la*
N	*just*
E	*saving*
R	*them*
D	*for*
S	*later*
}	*Stringtime is over!*
End of file	*Time to get to work.*

Figure 3: Paul Ford's narration of a computer's thought process for the subroutine `PRINT{HELLO NERDS}`

If the human process of coding appears tedious to the layperson, the computer's process of interpreting and acting upon code represents a "repetitive boredom that's unimaginably vast." Moreover, while markup projects such as the Text Encoding Initiative seek to describe with a degree of irreducible precision the aspects of textual entities and phenomena at the highest level, a computer seeks to simplify the code that it compiles and executes down to the lowest level of binary. This process is the focus of the specialized field of compilation, which seeks more efficient ways to abstract data so that the process takes less time and requires less computing resources.[22] Apropos of the digital humanities, Ford uses literature to explain what a computer does with code; if we begin by writing "elegant, high-level code like F. Scott Fitzgerald...the computer will compile you into Ernest Hemingway." But compilation is recursive, so the

[22] For reference, Emma Haruka Iwao set a new world record for calculating over 31 trillion digits of pi (π) in 2019, and this process, however simple its mathematical concept, required 25 Google Cloud virtual machines to run for almost 112 days (7.6 machine years), and used 170 terabytes of data storage space for a total input of 9 petabytes and output of 8 petabytes.

computer turns the Hemingway into "Stephen King, to Stephenie Meyer, all the way down to Dan Brown, each phase getting less readable and more repetitive as you go." From a computer's point of view, then, the data and phenomena that code represents are entirely "arbitrary" insofar as what matters is that they are translated—even *transformed*—into increasingly legible, yet increasingly emptied abstractions. Like Cypher, the computer doesn't even see the code. All it sees is zero and one, off and on.

At first glance, the perceptions and purposes of code appear notably divergent between humans and machines, and, in the service of educating a general readership, Ford musters a considerable effort to convince us that code and computers operate in an alien sphere. To his credit, a materialist framework indeed suggests that a computer's epistemology depends upon diametrically opposed physical states that function as the building blocks of its signifiers into which both elegance and crudity are abstracted: in other words, the on-off, yes-no binary system that governs memory storage and, one might say, knowledge itself for machines. Bypassing analysis or evaluation of the signifer's relationship to the signified—i.e., never taking into account what code purportedly represents and how it purports to represent ideas and phenomena—the computer translates and transforms signifiers into a base level that is germane to its ability to understand information and perform actions. Ford's description of computers' worldview adumbrates the major departures of thing theory and object-oriented ontology from traditional humanistic thought, casting aside what Ian Bogost memorably describes as "the sieve of humanity" in favor of a speculative realism interested in what it's like to be a (nonhuman) thing (*Alien Phenomenology* 3). By contrast, the TEI `<correspDesc>` task force approached a coding problem precisely for its implications regarding how a markup language models the

singular ambiguity and complexity of a textual phenomenon in relation to human experience. In the case of correspondence, code prompts a reflection on the means and impact of representation; in the case of a computer, code prompts compilation and output, the rendering of machine language and the concatenation of physical-technical actions. Apropos of Bogost, we see a distinction between *ideas* and *stuff*, both being possible yet divergent meanings for '*things*.'

Behind such a distinction, however, lies a shared epistemological structure governing both the modeling of `<correspDesc>` and the execution of `PRINT{HELLO NERDS}`. For markup like XML, this is known as the ordered hierarchy of content objects (OHCO), but in layperson's terms it simply means that ideas and things are understood as a nested structure of containers. Figure 4 below offers two of Ford's illustrations demonstrating the OHCO structure of `PRINT{HELLO NERDS}`.

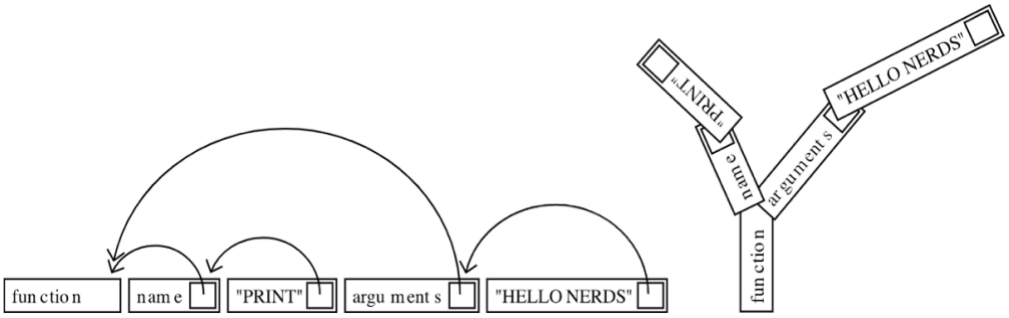


Figure 4: Ford's representations of the procedural structure of `PRINT{HELLO NERDS}`

For correspondence, the container that *describes*, i.e., `<correspDesc>`, contains any number of containers that specify *actions* and *context*, i.e., `<correspAction>` and `<correspContext>`, and within these containers are people, places, dates, and other information. "Every day is Arbor Day in Codeville," Ford quips, by which he means: the tree-like structure that governs all thought and action via code dictates that "[e]verything ultimately has to get down to things in little boxes

pointing to each other." This is the affordance and the constraint of code not just as a technical system, vocational skill, or way of executing functions and storing data, but also as a way of both representing and proposing how the world works.

The world, of course, is more complicated than any single hierarchy, and the relationship of container to contained serves more as an invitation to consider possibilities and alternatives as opposed to imposing strict limitations in the context of digital humanities pedagogy. For markup language, an approach called stand-off markup allows for "several hierarchically incompatible schemes" or "multiple hierarchical ambiguities" by storing the non-hierarchical markup in a separate location from the text, either elsewhere in the same file or in a different file altogether, using pointers and ranges to connect the stand-off markup with the original text span (TEI, "Linking"). The TEI P5 Guidelines provide ways of encoding stand-off markup in XML, and coders also can indicate alternative hierarchies by using 'empty' elements. These elements mark single points in the text, as opposed to enclosing spans of text. For example, `<pb/>` indicates a page beginning and `<note/>` indicates an annotation, while attributes such as `@corresp` and `@target` can link empty elements to markup elsewhere, as in `<note target="#annotation1"/>`. Figure 5 below highlights the overlapping hierarchy that would occur if `<pb/>` were not an empty element.

```
I could not recall the name of <rs type= "person" ref= "#janedoe1"> the
woman who had <pb>helped me so many years ago</rs>. I continued to muse
on the event until a voice disturbed my daydreaming…</pb>
```

Figure 5: an overlapping hierarchy in XML (green area) if <pb> *were not* an empty element

By contrast, the empty `<pb/>` tag solves the problem of a new page beginning mid-paragraph; were the page and paragraph tags both enclosing spans of text, they would overlap. Empty elements offer a pragmatic solution for projects that wish to note hierarchies of secondary importance. They also prompt consideration of the purpose of a project: namely, what it seeks to investigate and what is more important for that investigation (Cummings, "World of Difference" i71).

In addition to hierarchical structures, what's unique to code—apart from other systems of representation—is the degree of explicitness and precision required when one "model[s] some select slice of the world" with all of its ambiguity, complexity, contradiction, context, and excess "in the formal environment of a computer" (Kirschenbaum "Hello Worlds"). In Codeville, then, not only is every day Arbor Day, but it's also the case that *nothing ever goes without saying* when it comes to describing what things are, how they are related to each other, what about them matters, and how they can and should be manipulated for various purposes. Yet, this exhaustion enacts restriction, and herein lies perhaps the most provocative and productive aspect of code. In courses on new media and electronic literature, Matthew Kirschenbaum, for example, asks his students to model snowballs as an exercise in object-oriented programming:

> What are a snowball's salient characteristics? What do you do with one? Well, you toss the snowball at someone else. But wait, before you do that, you first have to shape it, form it, pack the snow. Once you do toss it, do you still have it? No. So the program has to be able to distinguish between possession and nonpossession of the snowball. And maybe, if you hang onto it too long, it starts to melt. The exercise of thinking through what it takes to model a snowball in a believable fashion goes a long way toward

capturing the appeal of what I mean by programming as world-making.[23] ("Hello Worlds")

The TEI Guidelines, essentially, are a large-scale effort to model snowballs for textual scholars. What are a document's salient characteristics, and how should they be encoded? What matters for recording instances of people, places, and other named entities in a text? How can the code provide for the excess of context without an unnecessarily baroque architecture or a prohibitively voluminous lexicon? "Programming is about choices and constraints," Kirschenbaum asserts, and these too are part of the radical transparency of code: the implicit made explicit ("Hello Worlds").

<(In)visible Algorithms and the Value of Coding Literacy/>

Nothing may go without saying in terms of the making-explicit of otherwise implicit attributes, processes, and rules, and yet, especially in the shadow of poststructuralism, we take as a given that many things *do* remain unsaid for any articulation, no matter how exhaustive or coherent the attempt. Put simply, behind the 'choices and constraints' slogan lies the assumption that code will always leave something out, perhaps a great many things, while its structure and

[23] As an activity for the digital scholarly editing course detailed in chapter four, I adapted Kirschenbaum's thought experiment by asking students to model correspondence (the type of document with which they work for the Breckinridge Correspondence and Digital Texts Project at the University of Kentucky). Using post-it notes, students brainstormed aspects of letters and correspondence: as act, as text, as document, as material object, and so on. As a collaborative affinity map exercise, the students then placed all notes on the wall and decided on how they'd like to consolidate, organize, and revise their model of correspondence by named traits and categorical and hierarchical relationships. For example, they discussed how a letter is sent, transferred, and received. They also discussed how a letter might include types of enclosures and other associated materials. I emphasized to students that this exercise approximates, albeit in an oversimplified form, how the TEI council deliberates on the TEI guidelines.

language will not align equally well with all contexts or cases. For Kirschenbaum, this uncodable excess represents the great "asset" of coding for humanities education via "sites of exploration, simulation, [and] play" ("Hello Worlds"). In deciding what to include, how to include it, and what to leave out when modeling concepts, objects, and other phenomena, students engage in the production of digital and algorithmic ontologies.

Behind these inquiries lie real stakes for understanding how code works in situations that are not designed as transparent, reflective learning experiences for the end user. Perhaps the most ubiquitous example is Google's search engine. In *Algorithms of Oppression*, Safiya Umoja Noble coins the eponymous term to describe how racist and sexist worldviews influence the ostensibly objective means by which an Internet search determines the most relevant results for a given set of terms. As noted in many reviews and press releases, the book opens notably (and disturbingly) with the pornographic top results for a Google search for "black girls" in the early 2010s. A 2012 update to Google's search engine suppressed pornography in the results for "black girls" (first in the top results only, and later that year for all results), but Noble remarks that "[i]t is impossible to know when and what influences proprietary algorithmic design, other than that human beings are designing them and that they are not up for public discussion, except as we engage in critique and protest" (4). In effect, then, we see through two glasses, darkly; beyond the opacity of the interface lies the opacity of trade secrets, commercial interest, and intellectual property.

In this case the apparent neutrality of the algorithm, the expansiveness of the database, and the brand-ethos of Google itself as the world's search engine elide the deliberate choices in locating, prioritizing, and manipulating data and metadata that model critical aspects of lived

experience. We are reminded that how search engines assign priority does not always align with traditional markers of expertise, credibility, and authority. Noah Wilson distinguishes this unique "algorithmic ethos" for its indices of virality—hits, links, likes, shares, reactions, comments, and so on—that constitute a populist rhetoric allowing for the circulation of otherwise incredible information and arguments (e.g., conspiracy theories).[24] Even further, Google's efforts to achieve real-time indices of the web and customize results to individual users' browsing data lead some to describe the search act, protean-like, as a posthuman rhetorical situation in which the algorithm acts upon the user, using its available means of persuasion (Graham, Zook, and Boulton 470). Wendy Chun explains this phenomenon as the "enduring ephemeral...of constantly disseminated and regenerated digital content" that, Graham, Zook, and Boulton argue, makes "a largely opaque contribution" to our augmented realities (148-49; 470). For Noble and other scholars, the stakes are high; not seeing the code becomes akin to not seeing race or gender: the rhetoric of erasure and invalidation with which people of color, women, LGBTQ people, and other vulnerable populations are all too familiar, and that tacitly authorizes violence and oppression in the name of civility and post-identity politics.

The problem of visibility manifests in the very concept of the search, given that it contranymically suggests both exhaustive perusal and selective attention (Gailey 125-26). As a basic example, a scholar may choose to read the entirety of a text to find particular ideas and references, or they may choose to use the Command+F (i.e., find) feature to read only parts of

[24] Social media activity surrounding the 2016 and 2020 U.S. Presidential elections has drawn increasing scrutiny to the algorithmic ethos of platforms from Google to Facebook. In an August 2020 interview for *The Daily*, the podcast arm of the *New York Times*, Twitter CEO Jack Dorsey expressed a degree of regret (of the 'we didn't know any better' variety) for how the platform's algorithmic ethos has incentivized sensationalism, especially for purportedly journalistic user accounts. Dorsey additionally speculates on a future in which search and social media companies openly provide their algorithms and even allow users to modify how they work individually (Jackson & Ibekwe).

the text that include specific words associated with the desired information. The Cmd+F algorithm is fairly straightforward; to extract particular information from a text, the interface directs readers to specific sections that match inputted words. But the method does not account for how ideas and information manifest in a text beyond exact language matches, as well as how the location and context of those matches colors their meanings. An additional layer of uncertainty emerges if the user searches for metadata in addition to the data (i.e., words) of a text; in this case, coders' decisions about how to apply the lexicon and structure of a coding language critically affect what the user sees and interprets. In other words, we must assess the validity of metadata, not solely in terms of accuracy, but also in terms of the arguments that it makes rather than take its objectivity as a given.

For projects involving TEI-compliant XML, this problem emerged for literary historian Amanda Gailey in her work as a co-editor of *Race and Children's Literature in the Gilded Age*. To encode texts in a way that made them searchable, Gailey and her co-editors needed to offer a version of the text that standardized the eye dialect endemic to historical representations of Black speech. Using the `<choice>` element in the TEI Guidelines, editors can encode language variants that may be shown or suppressed as part of the graphical user interface. (This is accomplished by using extensible stylesheet language transformations, or XSLT, which selects, arranges, and converts XML into other formats such as HTML.) Elements nested in `<choice>` designate the variants, with one pairing being `<sic>` and `<corr>`, the former designating original language as it was spoken or written (usually with a connotation of correct and incorrect usage) and the latter indicating the corrected language supplied by the editors. However, Gailey's team did not want to "make [any] claim about the rightness or wrongness of the readings"; rather, the team used

<orig> and <reg> to indicate language in its original and regularized forms, making a claim "only [for] how standardized their spellings are" (136), as in the following example from Joel Chandler Harris's "Uncle Remus" stories:

```
<choice>
  <sic type="eye-dialect">Dar once wuz a time when most er de creeturs</sic>
  <corr>There once was a time when most of the creatures</corr>
</choice>

<choice>
  <orig type="eye-dialect">Dar once wuz a time when most er de creeturs</orig>
  <reg>There once was a time when most of the creatures</reg>
</choice>
```

This particular case of <sic><corr> and <orig><reg> prompts a larger reflection on how the TEI and XML can model the complexity of a necessary editorial intervention into a white author's mediation of Black folklore that nevertheless cast a long shadow well into the twentieth century as one of the corpuses of record for the genre. It is unclear whether and to what degree Gailey and her team couched their deliberations on markup for Harris's eye dialect in editorial or cultural concerns, i.e., the editorial ethics of approaching what Gailey calls "heavy" interventions as more "correct" than the text itself, or the cultural ethics of suggesting that regularized speech text (in other words, the imagined ideal of Standard English) is more correct than dialect variants that mark racial difference. As a further complication, the variant in question cannot be taken at face value given that Harris's representation of speech does not seek ethnographic fidelity and is inextricable from the white supremacist logic that would distort Black speech as unintelligible and primitive, a proxy for overall intellect and personhood. While Gailey's concerns remain with the editorial and cultural framework of literary history, we may also trace her team's conundrum to the very OHCO structure of XML; because the sibling relationship between child elements of

<choice> suggest equivalency at the structural level, Gailey's team turned to the TEI lexicon to add shades of difference. The use of the attribute @type allows further distinctions, and an "eye-dialect" value added to the <orig> element could be complemented with a similar distinction for the regularized <reg> text with an attribute value of "machine-searchable" (thus clarifying: what does the regularization value? what is its attitude towards the original?):

```
<choice>
  <orig type="eye-dialect">Dar once wuz a time when most er de creeturs</orig>
  <reg type="machine-searchable">There once was a time when most of the creatures</reg>
</choice>
```

In any case, Gailey's team applied markup to make visible a neglected, yet influential corpus of writing, and they grappled with the visibility of the editorial interventions that undergirded the recovery and presentation of those texts. How code operates upon the text becomes as significant as the text itself. We engage, rather than avoid, the tensions between computational tractability and lived experience (Posner, "What's Next").[25]

Beyond end users and scholars, code may operate invisibly even among professionals and programmers in the IT industry. From a practical standpoint, how front-line programmers work with algorithms may preclude the explicit transparency that digital humanists find to be code's greatest asset. In "What Is Code?" Ford distinguishes algorithms from programming languages among other interrelated and often contested terms that often fall under the general aegis of code. Algorithms, Ford explains, are structured ways of solving problems, while programming

[25] Among other examples of the tensions between the tractability of data and the complexities of lived experience, Posner discusses *National Geographic*'s 2013 interactive essay "The Changing Faces of America," which invites readers to explore the differences between how people self-identify with respect to race and ethnicity and the Census categories that they must check (www.nationalgeographic.com/magazine/2013/10/changing-face-america/).

languages are “algorithm management system[s]” that allow for the “encoding, naming, and organizing [of] algorithms for reuse and application.” Rhetorical explanations of algorithms read similarly: e.g., “a procedural framework for accomplishing a particular task” and “the description of a task-oriented procedure through its component operations (i.e., its steps)” (Brock 33). Algorithms need not come from the computer age at all; as an ur-example, Ford cites Euclid’s algorithm, by which the greatest common divisor for integers may be identified, while Christopher Steiner, in *Automate This*, speculates that the oldest algorithm in surviving records concerns a Sumerian method for the equitable division of grain harvests without the use of scales (55). Steiner’s text—as popular nonfiction—gleefully channels a technological determinism most evident in the pronouncements of the final chapter, provocatively titled “the future belongs to the algorithms and their creators,” whom Steiner elevates as an intellectually superior oligarchy while suggesting that their algorithms will shape our lives increasingly independent from human decision-making (215). There’s a trace of social Darwinism bolstering the urgency of these sorts of pronouncements; *Program or Be Programmed*, warns the title of media theorist Douglas Rushkoff’s 2010 book-length manifesto, which admonishes us to “learn how to make the software, or risk becoming the software” (128). For critics following Rushkoff’s argument, becoming the software signifies a collective “stupefaction” as a result of merely using tools without understanding how they work: a familiar longue durée narrative of decline that echoes across literary classics such as *The Time Machine* as well as popular films such as *Idiocracy* and *WALL-E* (Bork 3-5). Our only option in such a techno(dys)topia, it seems, is to try to keep up and watch our backs in a “future where rigorous and cold calculations …constantly take place at the hands of algorithms” (Steiner 214).

Yet, as others remind us, algorithms first and foremost involve *human* hands. In an essay for *The Atlantic* Ian Bogost sets his sights on the metaphorical "cathedral of computation" that we buttress when concepts like "big data" and "algorithm" circulate discursively as mystical black boxes. In so many cases, Bogost argues, "the algorithmic metaphor gives us a distorted, theological view of computational action" while, given the proper scrutiny, "every algorithm betrays the myth of unitary simplicity and computational purity." The much-hyped 2009 Netflix Prize, for example, sought an algorithm that recommended films for individual customers, which resulted in the hyper-specificity of custom "altgenres" such as "Dark Independent Mother-Daughter Dramas." The seemingly singular accumulation of detail may indeed convey a sense of the Netflix algorithm's baffling, quasi-deific precision. Upon investigation, however, Bogost found the banality and inextricably human messiness of trained viewers who watch Netflix content and manually tag it with metadata, which then is used to name and populate altgenres for users (not unlike the accumulation of strategic details as one way of achieving a realism effect in literature). In the name of precision, Bogost defines the actual algorithm in play as the predictive matching of metadata between watched and unwatched content, while the larger Netflix project implicates "different systems, actors, and processes" of a global media company. The notion of Netflix engineers determining a metadata structure and lexicon for their content, with trained viewers assigning that metadata to audio-visual texts, has more in common with Gailey's work on *Race and Children's Literature in the Gilded Age* than it would at first glance. Amazon's job applicant recommendation boondoggle, too, presents a variation of this general approach if we view a supervisor's decision to hire or not to hire as a way of assigning rudimentary metadata to

job applicants and application materials, i.e., “desirable” or “not desirable” for given positions and types of labor.

Ford, Bogost, and others aim to remind us of the rhetoricity of code objects, particularly as they reproduce ideological, cultural, material, epistemological, and ontological frameworks. Drawing from computer science and critical media scholarship, Estee Beck similarly writes against the notion that algorithms are mere “agnostic language objects” that we may take as a given. Algorithms implicate our “values of logic and organization” and “how people...think and communicate” with an orientation to “a basic relationship of problem and solution.” Echoing Gailey’s work, Beck further “theorize[s that] computer algorithms are persuasive because of their performative nature and the cultural values and beliefs embedded/encoded in their lingual structures.” Attending to these values, a rhetorical orientation to code operates within a “scale of critical inquiry” that does not take the purpose and process of algorithms as a given, but rather as the most germane site for exploration. In some ways this scale resembles a “novice mentality” in its openness to ideas and arguments that a more trained coder may ignore in cleaving to standard or received approaches (Brock 52). As in the classic dendrological wisdom, they may miss the forest for the trees.

For a working programmer, then, coding does not necessarily involve the composition and analysis of algorithms per se as much as it involves their application and arrangement: a difference we might describe—or code—as heuristic purposes on the one hand and instrumental purposes on the other. This implicit division of labor along an intellectual hierarchy emerges in no small part from a historical tension in how the discipline of computer science has sought to define itself since the mid-twentieth century. As Annette Vee explains, programming challenged

academic and theoretical approaches with what was perceived as "dirty, practical, hands-on work," and it prompted the need to distinguish the academic field as a *science* as opposed to an art, all of which evokes the troubled history of programming as an under-professionalized and gendered labor category (12).[26] Reactions to the widely compatible and easy-to-learn COBOL (Common Business-Oriented Language) in the 1960s, for example, reveal a gatekeeper rhetoric among a new generation of male computer scientists who valued "abstruse knowledge" over the "rote, unintellectual, feminized work" of programming. Designed by a committee led by women, COBOL hardly advanced the male, neoliberal, lone-genius figure that has come to dominate the cultural imagination with respect to programming and digital technology initiatives. As gender and technology historian Mar Hicks remarks of this shift, "the last thing many male computer scientists entering the field wanted was to make the field easier to enter or code easier to read, which might undermine their claims to professional and 'scientific' expertise."

Distinctions of intellectual labor continue to shape the discourse on programming as a professional practice. Critics claim that in many cases a programmer's role on projects and in organizational hierarchies does not demand a sustained reflection on how formalized models and procedures convey assumptions, values, perspectives, and arguments beyond the instrumental needs of the task at hand. These sorts of programmers, coders, or software developers may focus more on *how to make the code work* within given parameters without venturing into questions

[26] Among other research, Beck cites Janet Abbate and William Aspray's *Recoding Gender*, which details metaphors for understanding computers during the mid-twentieth century: "as an engineering tool, a mathematical device, or a business machine." Based on these comparisons, "programming might appear high or low status, abstract or concrete, creative or routine," which aligned with assumptions about the skills that women brought to the workplace such as clerical work, using office machines, and accounting (53-54).

concerning *what it means* for the code to work (or not to work) in particular ways. Writing from his experience working in the industry, Ford reveals that

> [y]ou can do a ton of programming without actually thinking about algorithms—you can save something into a database or print a Web page by cutting and pasting code. But if you want the computer to, say, identify whether it's reading Spanish or Italian, you'll need to write a language-matching function. So in that sense, algorithms can be pure, mathematical entities as well as practical expressions of ideas on which you can place your grubby hands.

The imagery of grubby hands set against a mathematical purity (which in its own right evokes Bogost's cathedral metaphor) reifies the labor hierarchies that have become such a touchstone in popular and academic discourse on code, calling on—as if a subroutine—the archetypal figures of the physical mechanic and the intellectual engineer along with the assumptions, values, perspectives, and arguments that they implicate. In the digital humanities, too, we observe tropes of manual labor, as in Kirschenbaum's "code monkeys" or in Clive Thompson's blue-collar worker who "slings JavaScript," the echo intentional with how we often discuss the work of fast-food employees. Interestingly, in Thompson's piece, this code slinger works at a bank, where the web-based user interface—the primary application of JavaScript—may profoundly affect how a customer understands and controls their finances. The tasks may be (sub)routine, but they nevertheless demand a critical eye. Hicks has described this revaluation of maintenance and the everyday as a "labor of care" that rejects the neoliberal emphasis on disruptive, unsustainable, and inequitable innovation.

Contra analogies that figure code as car parts or fast food, Ford writes of algorithms as *expressions of ideas*, and his careful distinctions between ways of thinking about (and through) code bear particular urgency if we follow arguments for code as a universal literacy similar to writing. In fact, writing offers a rich comparison for its development as a core literacy in higher education, having expanded beyond the traditional and more pedigreed curriculum of scholarly and artistic approaches into more embedded and popular uses, e.g., writing across and in the disciplines, digital and multimodal composition, and professional and technical writing. One may predict a future in which code literacy and coding practices have similarly diffused throughout a spectrum of curricular exigencies.

Put simply, we're missing the point if we teach and learn *programming* without a focus on how code constrains and enables ways of thinking, communicating, and problem solving, or, in Vee's thinking, if we consider programming merely as a "specialized skill" as opposed to "a more common and literacy-like practice" (19). As Vee shows, this practice bears many names and descriptions in the literature (19). Michael Mateas has defined a "procedural literacy" as "the ability to read and write processes, to engage procedural representation and aesthetics, [and] to understand the interplay between the culturally-embedded practices of human meaning-making and technically-mediated processes" (101-2). Drawing primarily from video games, Ian Bogost explains "procedural rhetoric" as "the art of persuasion through rule-based representations and interactions rather than the spoken word, writing, images, or moving pictures," a rhetoric "tied to the core affordances of the computer" (*Persuasive Games* ix). Andrea DiSessa locates a notion of "computational literacy" well beyond the ability to use a computer (i.e., "computer literacy") as a larger, "infrastructural" shift that will "allow civilization to think and do things that will be

new to us" and that "improv[e] our abilities to represent the world, to remember and reason about it," all of which draws from a new "intelligence achieved cooperatively with external materials" (5). Among Jeannette Wing's litany of definitions for "computational thinking," we read that it is about "using heuristic reasoning to discover a solution" and "involves solving problems, designing systems, and understanding human behavior" (33-34). Lastly, Vee coined the term "proceduracy" in her early work for "the ability to break down complex processes into smaller procedures and express them explicitly enough to be read by a computer" (Hunter).[27]

More recently, Kevin Brock asserts the emergent field of "rhetorical code studies" as a way of framing and analyzing "software use and development as well as the communicative work that takes place through its code texts." He speculates that "such inquiry could open up new directions for pedagogical engagement with code and computation as avenues for communication as well as for critical literacy" (11). Brock's explicit engagement with code as a rhetorical field is innovative while his notions of critical code literacies echo Stuart Selber's foundational work on computer literacy. According to Selber, computer literacy attends to "the political, social, and even psychological assumptions embodied in computers" as well as how those assumptions are "instantiate[d]" among audiences of users. The "generative" potential of Selber's approach lies in the analysis not only of the functional potential of computer technology itself (i.e., as tool) but also of how the technology is used and how different kinds of use are shaped by, and in turn shape "larger cultural forces" (86-87). Toby Coley adds a dimension of "ethical literacy" to Selber's framework (13), which we observe in propositions that a computer-

[27] Vee's definition of *proceduracy* here aligns strikingly with Bernard Stiegler's notion of grammatization, both of which will be taken up at length in chapter three.

literate pedagogy should ask students "to analyze online environments, software, or hardware programs, asking who profits, who is left behind, and what values are embedded in these environments" (Sheffield). From literacy studies, digital humanities, and computer science, voices continue to distinguish the value of coding as a broad literacy and educational endeavor against competing claims that draw from an ethos of vocational training, economic pragmatism, and the technological zeitgeist. Using a framework of computer literacy, we see code less as a means to serve the information economy (of a particular nation-state, perhaps) and more as writing with, for, and through computers in ways that bolster understanding, contribute to critical discourses, and exert transformative agency in the world.

<Writing (and) Code/>

Scholars in writing studies have likewise navigated the tension between demands that others place on the discipline—usually prescriptive and instrumental—and the pedagogical aims of the discipline itself. As with writing, once a skill enters the pantheon of general literacies our scrutiny increases exponentially with respect to how we invoke it as a proxy for personhood and personal value, how we cite it to infer social developments or 'signs of the times,' and how we deploy it in service of worldviews both revolutionary and conservative. Deborah Brandt suggests that the more ostensibly democratized and de-specialized a literacy practice like writing becomes, the more it becomes an overloaded signifier supplying occasion or evidence for larger arguments or cultural narratives (26). Writing ability has become a synecdoche for a person's general intellect and socioeconomic value, for systemic problems in educational institutions, for

cultural and generational shifts, and more. So too with code, argues Vee, as languages, platforms, and devices proliferate with increasingly friendly user interfaces, and as code "tak[es] over the functions [traditionally] assigned to writing" (23). Overall, Vee finds, code has joined reading and writing in a foundation of "moral goodness and economic success" that signals "the health of a nation and its citizens" (3). In a public address aimed at students during Computer Science Education Week in 2013, President Barack Obama declared that "learning these [programming] skills isn't just important for your future; it's important for our country's future." Evoking the American maker archetype, Obama used code—and digital literacies broadly—as a stand-in for economic growth, national security, and global leadership, as well as a way of participating in the American exceptionalist narrative.

The use and ubiquity of digital platforms that Obama and other leaders imbue with *kairos* also invite speculation on generational shifts, as well as narratives of progress or decline. One of the most enduring theories for digital literacies in learning environments is Marc Prensky's notion of digital natives and digital immigrants. In the 2001 essay that coined the terms, Prensky writes that "[o]ur students today are all 'native speakers' of the digital language of computers, video games and the Internet," and he continues to rely on linguistic tropes with the notion of a digital immigrant "accent" that can be lighter or heavier, e.g., as indicated by asking an assistant to print an email rather than reading it on a web browser (1, 3). Prensky's theory has since been debunked as a caricature of sociology and cognitive science that trades in classist, ableist, ageist, and racist generalizations not the least evident in the native-immigrant analogy itself. Educational researchers have challenged the learning myths on which Prensky relies such as the so-called natives' ability to multitask (Kirschner & De Bruyckere 140). Yet, the distinction

nevertheless has passed into our cultural lexicon as a marker of generational difference (e.g., Millennials, Gen Z) and serves as a floating signifier for narratives of decline such as Jean Twenge's pop-psychology polemics *Generation Me: Why Today's Young Americans Are More Confident, Assertive, Entitled and More Miserable Than Ever Before* (2006) and *iGen: Why Today's Super-Connected Kids Are Growing Up Less Rebellious, More Tolerant, Less Happy and Completely Unprepared for Adulthood* (2017). In a way, however misguided, these sorts of provocations underscore the need for digital literacies, code foremost among them, to be considered alongside reading, writing, numeracy, and science literacy as indispensable for all learners and educational pathways.

For educational institutions, a literacy's foundational status is reflected in the curricular structure. Writing's long-naturalized place in core or general-education curricula communicates its value as a mass literacy spanning the disciplines, yet that very position may appear to free other courses, programs, and disciplines from addressing writing in a sustained manner. While commonplaces among writing studies scholars emphasize the need for embedded, iterative, and reflective writing practices, educators in other fields—especially those with a rigid prerequisite curricular approach that assumes knowledge, skills, and methods ostensibly acquired in previous courses—may see in the core writing requirement a promise of technical and methodological mastery that (ideally) may be called upon with unimpeded replicability. However, research has shown a need for continuing mentorship and practice regarding how writing is embedded in different genres, disciplines, and contexts (Goldschmidt 36). As writing across the curriculum (WAC) programs and practitioners remind us, all writing is situated and all rhetoric epistemic. WAC scholars have persuasively established that writing serves learning best when it is iterative

and scaffolded throughout coursework with explicit attention to the differences and similarities with previous writing tasks (Croft, et al. 194-95). We find the same argument throughout the *Framework for Success in Postsecondary Writing*, a 2011 report published by the Council of Writing Program Administrators, the National Council of Teachers of English, and the National Writing Project. In particular, the report asserts that "a writer's grasp of conventions in one context (such as a lab report for a chemistry class) does not mean a firm grasp in another (such as an analytical essay for a history course)" (9). These arguments convey a specific understanding of what writing *does* for students. Like any literacy, writing is not only a prerequisite to acquiring knowledge, nor just the evidence that knowledge has been acquired; rather, writing is a medium in which discovery and knowledge-making continue to be performed. Arguing for an expanded vision for knowledge production and assessment in the humanities, Stephen Ramsay emphasizes how "com[ing] to understand...an issue or a problem through the act of writing about it forms the basic pedagogy of the humanities." Oversimplified approaches to writing-based assessment, on the other hand, place writing *after* learning as mere proof that students "possess" some body of knowledge or can enact some kind of methodology. Ramsay would remind us that "[w]e assign writing...to have that thinking occur in the first place" ("Developing Things").

Outcomes for writing across the curriculum programs emphasize a "complementary, even synergistic" relationship between "writing to learn and writing to communicate," or, in other words, writing as both a heuristic, formative process and writing as a reportorial, summative process (McCloud and Miraglia 5). Discussions of learning outcomes for writing pedagogy must also contend with the protean notion of *critical thinking* as it bolsters the core function of writing curricula while also vexing our attempts to define just exactly what students are thinking about,

and how that thinking is different—and more valuable—than other kinds of thinking. For a better understanding of critical thinking in the context of code, digital writing, and composition I find it helpful to turn to geographic information system (GIS) scholars Sarah Elwood and Matthew Wilson. They describe a general goal for their GIS courses as "the critical," which involves:

> a way of thinking and doing spatial data and technologies that are robustly engaged in building students' technological capabilities and constantly foregrounding questions about how we know through the digital and spatial, where these forms of knowing come from, and what the consequences are. (5)

Elwood and Wilson respond to the marginalization of critical methods and perspectives in GIS programs; the title of their article announces a pedagogy "beyond 'Week 10: Ethics' " as an example of the typical curricular footnoting of critical approaches in more technical disciplines. Information and computer science researchers, similarly, have advocated for "more than just an ethics requirement for CS majors" in favor of a critical framework that would transform all aspects of computer science education (Ko, et. al). Elwood, Wilson, and others argue against a backdrop of expectations—those held by students, employers, policymakers, administrators, and other stakeholders—that favor instrumental competencies and technical mastery in programming languages and platforms. We see a similar push for digital literacies that aspire beyond "mere functional approaches" in writing program administration, a stance reflected explicitly in the title of Jenna Pack Sheffield's webtext "Thinking Beyond Tools." Surveying program administrators, Sheffield found that even though WPAs are unlikely to rank functional or technical literacies as a top priority for their program outcomes, "some of their discourses and practices still represent digital literacy in a functional, tool-based manner." In the case of writing, we often encounter

this rhetoric of instrumentality and functionality, from expectations that students know how to write research papers on command (as if all research writing were interchangeable) to the alleged decline of mechanical proficiency and stylistic sophistication in students' writing (which, most prominently, Andrea Lunsford has contested).[28]

With this focus on "the critical" as a way of balancing technical mastery with reflective and ethical inquiries, coding and writing pedagogy form a Venn diagram of interrelated learning outcomes. The similarities in language are telling; while Matthew Kirschenbaum has asserted that "[p]rogramming is about choices and constraints," John Warner has proposed that "[t]o write is make choices, word by word, sentence by sentence, paragraph by paragraph. Writers choose what they want to write about, whom they want to write to, and why they're writing" (5). To see the code is, in a way, to see the choices, and to be explicit about the rationales behind the choices that one makes. As Vee explains,

> [t]o write code, a person must be able to express a process in hyper-explicit terms and procedures that can be evaluated by recourse to explicit logic rules. To read code, a person must be able to translate those hyper-explicit directions into a working model of what the computer is doing. (*Coding Literacy* 22)

From the beginning, the pedagogical project that this dissertation represents has drawn from the notion that code in the humanities class—whether first-year composition or digital scholarly editing—uniquely makes visible and explicit for students the formal processes of the learning outcomes in terms of both the specialized goals (e.g., making genre-informed writing choices,

[28] As Lunsford explains in her touchstone longitudinal study of student writers, the average amount of writing per student and the complexity of assigned writing tasks have increased over time, while the errors per word count have remained the same. Those errors may differ in type, however, for which we might venture various speculations.

applying a coding scheme to represent historical correspondence) and the generalized target of *the critical* as a way of evaluating and intervening in complex scenarios. In the inaugural edition of *Debates in the Digital Humanities*, Alan Liu advocates for more intentional cultural criticism in computationally driven projects, and scholars have recently found a persisting need to apply critical methodologies in more digital humanities work. Some of the more opportune moments for a critical approach, they suggest, are the collection, organization, and description of data *before* analyses and interpretations—those default learning activities of the humanities—are performed (Long and Baker). Indeed, as Chris Aaron Lindgren finds from his ethnographic observations of a programming team, coding involves "decisions about what contextual and historical factors matter" and implicates "the role of the often-elided data-processing work to contextualize the data" to the point at which we may describe coding as "rewriting data for new goals and purposes" (148). The case studies in later chapters of this dissertation use markup language as a way of involving students in this critical work with (textual) data, which in turn attunes them to more reflective learning practices.

<"To See and Alter": Code and Critical Pedagogies>

Often reduced to vague shorthand like critical thinking or critical analysis, the notion of the critical as I have developed it here always risks falling into the pablum of general transferable skills. However, an intentional understanding of critical engagement with code is important because in many cases *technology does not want to be seen*. To teach students to see code and make visible the models, assumptions, perspectives, and claims that undergird the designs of

technical and technological objects and systems is to position them as counteragents where the presence and operation of technologies are supposed to continue without notice or scrutiny. From the earnest efforts of usability design to the limitations imposed by intellectual property, and finally to the more deceptive elisions and obfuscations of the reach and capabilities of hardware and software in a global technocapitalist oligarchy, devices and programs often do not seek to draw attention to (nor invite analysis of) how they work. To see the code is to participate in a larger pedagogical effort that (a) invites us to articulate the affordances and limitations of design elements aligned with dominant socioeconomic and ideological forces, and (b) fosters both the critical perspectives and technical skills to leverage languages, platforms, and devices in ways beyond their intended uses towards the creation of different forms of knowing and living. Following this understanding, *seeing* aligns less with the passivity of *watching* and more with the active, transformative work of *envisioning*: naming and moving toward an unrealized potential. This imaginative agency is more radical than it may seem at first glance given the degree to which code and information technologies seek to replicate—rather than transform—extant knowledge and perspectives. As Benjamin Schmidt warns, "algorithms will approximate existing meanings, [which] in many ways precludes them from creating new ones." (Recall the Amazon hiring algorithm that drew only from data generated from previous applications and hiring decisions.) Algorithmic action conforms to observable patterns, taking as a given that they are desirable as-is instead of evaluating them and enacting structural changes where they're needed. Schmidt explicitly draws from Ramsay, whose notions of text analysis and algorithmic criticism

urge us toward "the unfolding of interpretative *possibilities*" based on the machine analysis of texts (10, emphasis mine).[29]

In terms of *The Matrix*—which loosely provides the conceit for this chapter's vernacular theory of seeing the code—we seek to empower students to free themselves from and dismantle their virtual prison as consumers of technology, opposite Cypher's rejection of critical awareness and the burden of knowledge that he has carried. In a later scene that invokes consumption in its physical and figurative registers, Cypher arranges for the conditions of his reintegration and memory wipe as payment for his betrayal:

> You know, I know this steak doesn't exist. I know that when I put it in my mouth, the Matrix is telling my brain that it is juicy and delicious. After nine years, you know what I realize? Ignorance is bliss.

Tropes of liberation and incarceration are not out of character given their frequent occurrences in the literature on teaching and learning as it engages the technological and the political. Ramsay, for instance, finds in his vision for algorithmic criticism a potential "revolution" for text analysis that is "both welcome and liberating" (17). Yet, where we might break free we also might remain imprisoned. In their touchstone article on the human-computer interface, Cynthia Selfe and Richard Selfe have emphasized "the ways in which the borders evident in computer interfaces can be mapped as complex political landscapes" that reify containment protocols along

[29] Also drawing from Ramsay's work on the speculative potential of algorithmic criticism, Kari Kraus sets out to define her notion of "conjectural criticism" for computer-assisted textual scholarship. She describes the object of conjectural criticism as "a form of subjunctive knowledge" that is "notional rather than empirical; possible rather than demonstrable; counterfactual rather than real." Overall, her wide-ranging and novel analysis (e.g., arguing for nineteenth-century textual criticism's methodological influence on computational evolutionary biology) affirms an empiricism that does not merely seek to account for *what is*, but that also is committed to reflecting on what could have been and what still might be.

socioeconomic, racial, gendered, national, and linguistic lines (482). Their work has influenced scholars who continue to explore how computers and computation, even and especially in their most ostensibly objective or banal expressions, implicate power asymmetries. Echoing Selfe and Selfe's thesis, Wendy Chun has argued in her 2011 monograph *Programmed Visions* that "GUIs [graphical user interfaces] are a functional analog to ideology," pointedly adding that computers "are ideology machines" that reinscribe borders and hierarchies (66). In her extended critique of software and neoliberalism, Chun reflects on the contranymical meaning of the "screen" as (a) the apparent visibility and transparent control provided by the graphical user interface, and (b) the concealment of design choices and user management techniques: "a paradoxical combination of visibility and invisibility" (59).

Recent scholarship has emphasized the importance of code, albeit less visible than the graphical interface itself, as a digital infrastructure with the potential both to reinscribe or to deconstruct ideological narratives. As a case study, Brandee Easter's analysis of the 2013 C+= hoax (pronounced "C plus equality") shows how code can be weaponized for misogynistic discourses online. Presenting itself as an earnest attempt to imagine a feminist programming language, C+= was in fact the work of (male) 4chan users who sought to ridicule an academic blogger as well as women's participation in programming at large.[30] The commentary embedded in the code enacts a sexist and violent rhetoric, but Easter further explains how the code itself performs a "digital manspreading" with its structure and execution (676). The sheer volume of needless and sexist commentary embedded in C+='s version of the "Hello World" program

[30] For the original blog post targeted by C+=, see Ari Schlesigner, "A Feminist & A Programmer," Humanities, Arts, Science, and Technology Alliance and Collaboratory (HASTAC) Blog, 13 Dec. 2013, www.hastac.org/blogs/ari- schlesinger/2013/12/13/feminist-programmer.

extends it to 89 lines, hyperbolically dwarfing what actually is needed for the program to compile. In C++, even with extra spaces, “Hello World” occupies only seven lines. Moreover, as it is written, the C+= “Hello World” program would fail to compile precisely because it would use too much physical memory space. As Easter explains, a computer reserves a finite amount of memory to run a program, and when this space is exceeded, the program crashes. The C+= “Hello World” code contains an infinite recursion with a function called `PrivilegeCheck()` that calls on itself in a nested doll format. Rather than indicate a stack overflow error—the common reporting language for this sort of crash—the console would display that the program has failed to compile because of a “trigger warning” (682).

C+= does not represent an earnest effort to create a functional coding language so much as it leverages code as a medium and idiolect to reify the structural misogyny that so profoundly affects programming and information technologies. Ari Schlesginer, the target of the 4chan trolls, had attempted to begin a conversation about “how we can combine feminism and programming languages in a way that makes social responsibility, equality, and accessibility an effective part of the programming process.” Subsequent efforts have responded to this question, such as the Feminist.AI group’s Feminist Search project, which seeks “an alternative to private search engines” (Google foremost among them) and builds from the work of scholars such as Safiya Umoja Noble towards

> a visual search engine powered by community definitions and informed by library science and critical theory. With a belief in users as not only contributors, but also as owners of their experience and information, an editable, co-created search tool will be

> developed where users have control over access to their information and how it is used for development. (Meinders, Thompson, Ciston, & Griffiths)[31]

Feminist Search reimagines the search engine from a critical design perspective that values user agency, data ownership, semantic ambiguity, and decentered authority. It represents what Jeremy Douglass, along with the 2014 Critical Code Studies Working Group, has defined as *feminist code*: "executable, syntactically subversive" new coding languages that evince "a fundamentally different feminist paradigm for software development" (in Schlesigner, "Feminist Programming Language").[32] From the interface to the code that powers it, we note borders and lines—as well as occupations and schisms—that constitute the political landscapes to which Selfe and Selfe draw our attention. They underscore "the ways in which teachers and students can learn *to see and alter* such borders in productive ways" as part of what they call "a radical pedagogy of electronic borders and borderlands" (483, emphasis mine). Drawing from Mary Louise Pratt's foundational work on contact zones, Selfe and Selfe characterize these borders as sites of power imbalance and struggle, wherein lies the potential for both oppression and transformation. Other scholars frame the interface similarly, "as the newest layer of cultural oppression weighing down a human consciousness already burdened by the overarching nexus of power that subjugates the individual to society," which we can counter with a "digital humanities pedagogy of resistance"

[31] For a more detailed explanation of Feminist Search, as well as the code itself, see Christine Meinders, "Week 3: Feminist Search (Code Critique)," Critical Code Studies Working Group 2020, 3 Feb. 2020, wg20.criticalcode studies.com/index.php?p=/discussion/88/week-3-feminist-search-code-critique. For contemporaneous examples of feminist approaches to data and programming, see Catherine D'Ignazio and Lauren F. Klein, *Data Feminism*, MIT Press, 2020, data-feminism.mitpress.mit.edu/ and Sarah Ciston, "Imagining Intersectional AI," Conference on Computation, Communication, Aesthetics & X, 2019, 2019.xcoax.org/pdf/xCoAx2019-Ciston.pdf.

[32] Douglass distinguishes feminist code from *code feminism* and *feminist codwork*: respectively, the use of existing (i.e., non-feminist) code paradigms to accomplish feminist aims with executable/functional code, and "pseudocode" that cannot be executed but uses code ideolects or "code-like texts articulating feminist ideas or subject positions" (in Schlesinger, "Feminist Programming Language").

that bends technologies and traditions for methods and purposes beyond their prescribed design (Lecky 371; Wharton, "Bend" 388).

This pedagogical ethos of digital subversion must walk a tight rope by (a) making space for students to exert agency and challenge hegemonic narratives, while (b) ensuring that students still benefit from instructional guidance, support, and expertise. In other words, a liberationist approach to digital humanities pedagogy seeks to restructure power in learning spaces but also recognizes that we cannot place the burden of critiquing and transforming structures of power on those with the least privilege and access. Especially in the context of digital pedagogies and new media, instructors and other educational authorities can lead by example—and by design—when it comes to the interrogation of otherwise tacit assumptions and impositions of technological use and value. In William Kurlinkus's writing on design, nostalgia, and new media, we encounter an approximate idea in his explanation of "informed dissent (as opposed to informed consent)," whereby designers not only allow for the theoretical possibility of resistance but intentionally lead audiences to explore "complications and changes" that a design intervention might present for a target community. Like Kurlinkus's designer, the (digital) humanities instructor—who, knowingly or not, also works as an instructional designer—seeks to support students as "critical user audiences that can actively participate in democracy" and, I would add, in the classroom (101). Seeing the code, then, cannot rely on a one-way vector from instructor to student, but rather must proceed from an instructional design and pedagogical praxis that values students' reflective agency as participant-designers and active learners.

Like a drone note, the Freirean vision for an education that provides the tools and support for students to dismantle and transform the systems that oppress them undergirds and amplifies

pedagogical theories and practices in the digital humanities. One of the more vocal proponents of the critical pedagogical tradition when it comes to encounters with information technologies, Jesse Stommel defines a critical *digital* pedagogy with aims of both drawing from and creating new "communit[ies] and collaboration[s]" in ways attuned to the need for more than goodwill as the (infra)structural condition for "reimagin[ing] the ways that communication and collaboration happen across cultural and political boundaries." Should the critical educator venture to build such a structure, it will "gather together a cacophony of voices" in ways that do not explicitly or implicitly favor one or a subset as the speakers of a master discourse. Lastly, Stommel's critical digital pedagogy leads students to "use and appl[y their learning] outside traditional institutions of education" ("Critical Digital Pedagogy"). In general, appeals to use-value and applicability run the most risk of courting a sort of confirmation bias, whereby one assumes a rationale aligned with what one values most as the purpose of an education and the educational system. This is, put simply, the moment when people are most likely to hear what they want to hear. Importantly, what Stommel and others have in mind for application outside the classroom is not a skill set for career viability but rather a framework for re-visioning and revising the structural conditions for equity. Henry Giroux, a giant of the critical pedagogy movement, affirms the conviction that education ought to expand students' civic and moral imaginations to prepare them to "hold power accountable" in a life of public engagement. Such a lofty goal faces a good deal of inertia, however, stemming from "a culture drowning in a relentless love affair with instrumental rationality" and an "education [that] is reduced to either a private affair or a kind of *algorithmic mode of regulation* in which everything is reduced to a desired outcome" (28, emphasis mine). Writing as the United States braced itself for the 2016 presidential election,

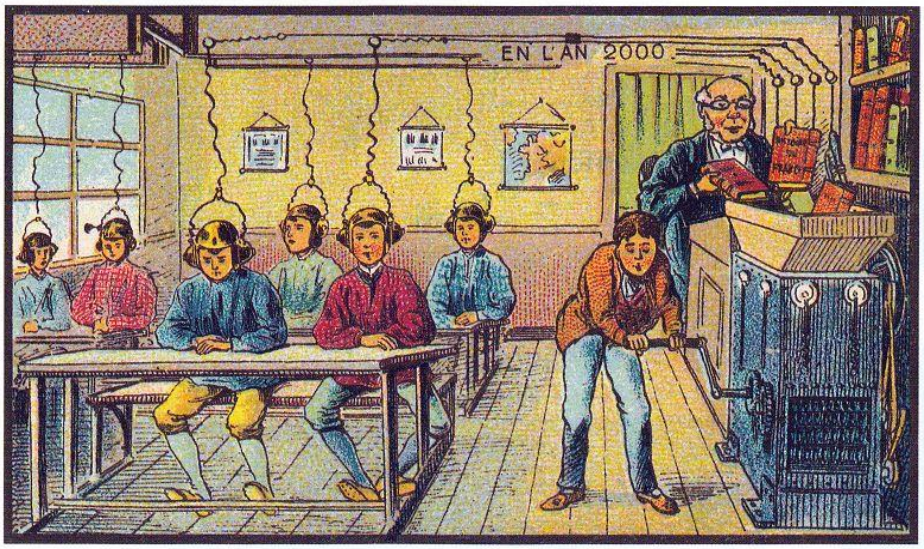
Giroux saw how biopower and technologies of control continued to thwart liberationist pedagogies in the post-information digital age.

Specifically in the digital humanities class, Stommel's critical digital pedagogy manifests as a sustained unveiling, interrogation, and alternative modeling of our technological *habitus:* how digital infrastructures shape our senses of self, others, and the environments we occupy, all of which produce and limit the possibilities of communicating and acting. We find a compelling example from Chris Gilliard, an English professor at a two-year college whom Stommel cites in his reflections on teaching during the COVID-19 pandemic.[33] Leading students to consider the physical and algorithmic infrastructures governing their access to networks and information, Gilliard has found that they "are often surprised (and even angered) to learn the degree to which they are digitally redlined, surveilled, and profiled on the web, and to find out that educational systems"—perhaps including their own institution—"are looking to replicate many of those worst practices in the name of 'efficiency,' 'engagement,' or 'improved outcomes.' " The driving question of possibility and liberation for Gilliard's pedagogical praxis is: "what would the web look like if surveillance capitalism, information asymmetry, and digital redlining were not at the root of most of what students do online?" That Gilliard holds this kind of learning space at a community college is significant in the context of critical pedagogies that would call attention to structures of power and privilege. Because most digital humanities centers, faculty, projects, and coursework are concentrated among larger institutions with established graduate programs and prolific research output, students at two-year colleges are most likely to be excluded from critical

[33] See "Designing for Care: Inclusive Pedagogies for Online Learning," 19 June 2020, www.jessestommel.com/designing-for-care/.

discourses that would otherwise take on new urgency precisely where transformative pedagogy and instruction are the raisons d'être (McGrail).

Underlying the arguments of critical educators such as Stommel and Gilliard is the notion that not only does technology not want to be seen, but, following that surreptitiousness, it likely does not work towards our best interests. As a historical example of this kind of phenomenon, Figure 6 below shows an entry in Jean-Marc Côté, et al.'s turn-of-the-century series *En l'An 2000*, published in an obscure collection by Isaac Asimov in 1986 but recently popularized by the open access website *Public Domain Review*.[34]



At School

Figure 6: "At School," a postcard image from Jean-Marc Côté, et al.'s series *En l'An 2000*

After noting what appears to be an all-male classroom, observers of this image tend to reach a conclusion that while it may have been composed in the spirit of technological utopianism at the

[34] See "A 19th-Century Vision of the Year 2000," *The Public Domain Review*, publicdomainreview.org/collection/a-19th-century-vision-of-the-year-2000.

turn of the twentieth century, it strikes us now as a kind of unintentionally prescient *dystopian* representation of educational technologies leveraged to precisely the opposite effect of Frierean empowerment. Indeed, the image seems to celebrate the great innovations (re)structuring life in the first decade of the twentieth century: electricity, the telegraph, and industrial mechanization, standardization, and scalability. The educational environment we glimpse is efficient, structured, and uniform. Yet, viewers are quick to point out aspects that clash with our own commonplaces of teaching and learning: a one-way vector from source (not even instructor) to student; a lack of engagement and community; an apparent ethos of conformity, discipline, and hierarchy; and the implicit pedagogical stance that equates learning with the transfer of information.

Beyond the fish-in-a-barrel exercise of critiquing the problematic futurisms of the past, many observers need no prompt to consider that we may very well continue to operate within what otherwise might strike us as a paradigm from a less enlightened time. Consider how "At School" anticipates the analogy of learning-as-downloading, literalized as how *The Matrix*'s Neo, in what has become a meme, comes to "know kung fu." Where technologies are the most enmeshed with our educational endeavors (hybrid and online learning, for one, though also the day-to-day operation of a college campus), we might witness an uncomfortable similarity with the Taylorist fever dream depicted in *En l'An 2000*'s "At School." Sean Michael Morris, co-founder of the Digital Pedagogy Lab and self-identified critical instructional designer, warns us about the "truly banal solutions" touted by university administrations, which would "reduce the complexity of [online] learning to straightforward methodologies that provide replicable results," all of which responds to "a perceived need for efficiency" with "a set of best practices, tools, [and] interfaces." In other words, the information technologies with which we teach—whether

the learning management system, student response system, website publishing platform, collaboration apps, or composing software; whether we're teaching online, hybrid, or in-person classes—represent opportunities for critical inquiry rather than the seamless integration of purportedly value-neutral tools and solutions for learning challenges that have been abstracted from their specific, complex, and intersectional contexts. Like Selfe and Selfe, Morris interrogates the digital interface as a site of power with the potential to limit or to expand students' literacies and agencies, all of which depend on how instructors design and facilitate the learning environment. These concerns have taken on new urgency in the wake of the global COVID-19 crisis and the exponential increase of educational technologies that facilitate hybrid, flexible, and remote learning environments.[35]

<The Slowness of Seeing and the Subversion of Value/>

The attempt to *see* code resists the techno-capitalist values of invisibility, seamlessness, and efficiency of labor and production, especially that which makes use of digital platforms and processes. In a touchstone treatise outlining a philosophy of "slow" technology, Lars Hallnäs and Johan Redström write that the convergence of ever-improving computational speed with the

[35] Educational leaders have described the challenges that Morris articulates as "wicked problems" that cannot be understood as "tame and solvable" nor in "bounded isolation" (Bass). In response, educators and students must be "intellectually nimble, capable of pivoting quickly to different methodologies, different ways of thinking, different concepts and modes of problem solving" (Hanstedt). Learning management systems, videoconference platforms, and remote proctoring solutions all have drawn considerable critique and in some cases outrage over their (failed) use as purportedly straightforward solutions to plug the gaps left by emergency remote and hybrid instruction. Plagiarism detection programs, too, fall into this category; for a representative critique of this kind of platform, see Morris and Stommel, "A Guide for Resisting Edtech: The Case Against TurnItIn," *Hybrid Pedagogy*, 15 June 2017, hybridpedagogy.org/resisting-edtech/.

omnipresent logic of productivity has led to the sacrosanctification of ease of use and immediacy of results, which tacitly authorize hidden automation and opaque processes as the other side of the coin. Quite often, this is in fact the stated desire for instructors looking to integrate innovative technologies into their teaching: something that requires minimal effort to learn and that doesn't distract from the subject matter or lesson at hand. In other words, we want a technology that quietly works and doesn't make its presence felt. As Hallnäs and Redström acknowledge, the bugbear of user-unfriendly, glitchy, and poorly designed technologies often precludes an openness to critical engagement with technologies from a purely affective domain. There's nothing quite like fiddling with a computer while students impatiently stare and lose interest, or planning a class meeting around the use of a program that doesn't load properly, to convince an instructor that the felt presence of technologies—the visibility of their operations—only serves to interrupt the learning that would otherwise take place. With an exponential increase in our use of technological platforms to continue delivering courses in the wake of COVID-19, instructors feel all the more keenly the need to spend time on course content and objectives as opposed to troubleshooting technical issues. We just want things to work, and we often don't care how, even in an educational mise-en-scene where we would otherwise encourage the careful analysis of assumptions, values, and arguments. Miriam Posner reminds us that the "prized" aspects of the user interface such as "transparency, seamlessness, and flow, privilege ease of use ahead of any kind of critical engagement (even, perhaps, struggle) with the material at hand" ("What's Next").[36] Considered as a means to accomplish some 'other' learning goal, digital infrastructures

[36] Posner's evocation of transparency may be confusing here given that I have otherwise written of transparency as an antidote of sorts to the obfuscation of the digital interface. In the passage quoted here I understand Posner's use of the term to signify a minimalist design approach that removes/obscures as much as possible from the user in the name of reducing cognitive load and funnelling actions to desired outcomes (e.g., Amazon's one-click order button).

remain unimportant beyond their instrumentality, and their operations remain important only for subject areas that deliberately study them such as engineering, data science, and web design.

As an imaginative contrast, Hallnäs and Redström's utopian vision of "slow technology" reinscribes the visibility of all technologies, including code, so that citizen-users may reflect on how the technology works, why it has been designed to work in such a way, and how it impacts the world. Moreover, this reflection does not settle with passive contemplation and encourages user experimentation and manipulation unbound by the purposes for which the technologies were created (cf. Wharton's bending of technologies and Kurlinkus's informed dissent). Ultimately, Hallnäs and Redström explain, slow technology "is concerned with amplifying the presence of things to make them into something more than efficient tools for specific, well-defined tasks," and, secondly, with "exposing technology in a way that encourages people to reflect and think about it" (203). Their examples may strike a reader as far-fetched, such as a doorbell that rings each instance with progressively more complete passages of a tune at a slower rate than would be expected. As unabashedly utopian provocateurs who stand at the intersection of art, design, and engineering, Hallnäs and Redström mean to shake us from assumptions about how technologies ought to work, and what the work of technologies is about.

Similarly, writing instructors may seek a "defamiliarizing" effect when asking students to compose in new forms, genres, and modalities, such as revising a written assignment into a video essay. Using this approach for an assignment sequence on science writing, Michael Ennis found that the less familiar aspects of video production and editing led to critical reflections in which "students commented on how they had taken their writing choices for granted, and creating the video prompted them to consider those design choices more carefully" (49). As users who are

inundated with digital interfaces, we, too, risk naturalizing the assumptions and frameworks that undergird how we receive and produce information. Drawing our attention to this risk, Bethany Monea presents annotated images of modified GUIs with the stated goal "to help us *see*—and *see beyond*—the limits of the interfaces that mediate our everyday digital lives" (emphasis added). We are indeed arrested and shaken from the familiar when we click through Monea's examples: an upside-down Google Map of the world, a slideshow of search portal interfaces from cultures and regions around the world, a document interface in which all functions draw from garden metaphors as opposed to the office metaphors that have always informed the language of word processors, etc. Figure 7 below shows a screenshot of Monea's garden-based GUI, which invokes garden iconography and language such as "sow" on clickable buttons while displaying plans for a garden from a bird's-eye view in place of the 'page' in word processors.

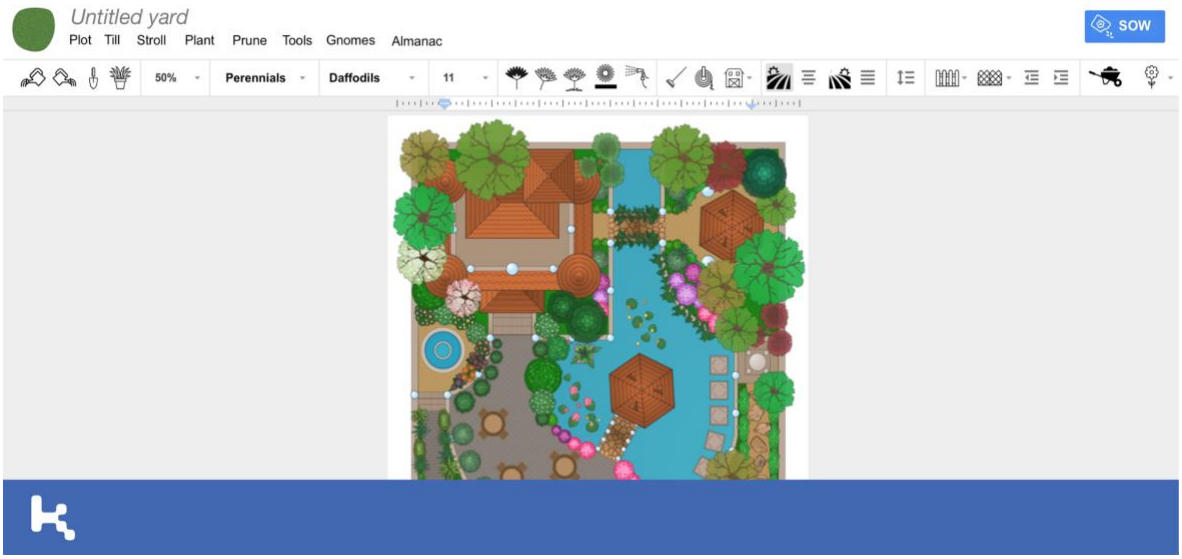


Figure 7: detail of Monea's modified GUI for a word processor using garden-based skeuomorphs

Recalling Bogost's argument about algorithms' embeddedness in economic, social, political, and material networks, Monea also includes a slideshow in which images progress through an

Amazon product webpage, the code that generates the page, the hardware that runs the code on the user's device, the servers that host the code, and the physical facilities and natural resources that maintain and power the servers. From Hallnäs and Redström, Ennis, and Monea's work we may infer the radical potential of seeing. By making a technological medium conspicuous, we invite readers and composers to reflect on how the medium, code or otherwise, exerts persuasive agency. This invitation resists the invisibility and speed that have become the hallmarks of our digitally mediated and lived experiences of the world.

As we continue to generate data at a geometric rate, the temptation to yield our care and control to computation is strong indeed. Only code and algorithms can parse, manipulate, and filter that data while maintaining a pure affective indifference to the "mundane processes" that they perform like clockwork, as in Ford's notion of second-grade math repeated ad infinitum (Hickman). On the one hand, then, it seems that, however imperfect and intimidating, code and algorithms represent the only tools we have for understanding the oceans of data in which we all seem to be swimming. Andrew McAfee, an MIT research scientist who studies the digital economy, makes perhaps the most provocative argument along these lines. "[A]s the amount of data goes up, the importance of human judgment should go down," he writes, even when that judgment is informed by the very computational analysis that he elevates. By contrast, others warn that we must not let our sense of incapacity and overwhelmedness authorize an information "dictatorship" or data "fetish" that discourages critical perspectives and, at best, leaves room for the misuse of seemingly objective calculations (Mayer-Schönberger and Cukier 151).

'Big data' has become shorthand not only for a quantity of information, and not only for the need to develop innovative ways of analyzing that quantity and training programmers and

data scientists (e.g., recent growth of degree and certificate programs, as well as industry-based boot camps), but it also has become shorthand for a tacit argument about how we should make sense of ourselves, others, and the world. Big data, in other words, proposes a worldview. The potential for large-scale analysis is undeniably attractive, perhaps leading to breakthrough ideas about systems and nature that lie beyond our cognitive faculties.[37] At the same time, that exciting potential unfurls with and within political economies, market relationships, and sociopolitical processes. "In this context, scale is an end unto itself," writes Tressie McMillan Cottom, who finds that this tacit value proposition is evident in digital humanists' undertheorized use of social science methodologies.[38] For instance, in Matthew Jockers's *Macroanalysis* (a touchstone work following the 'distant reading' school of Franco Moretti and the Stanford Literary Lab), we observe what Moher, Burgess, and Menzies describe as the undertheorized dualism of "human fragility [and] numerical certitude" (190). Making a case for the persuasiveness of macroanalysis and distant reading, Jockers describes the hypothetical project of comparing the writing styles of Hemingway and Joyce. According to Jockers, a "computational approach would be *all the more convincing* for being both comprehensive and definitive," while a comparative close reading of selections of canonical texts would be, much less convincingly, "anecdotal and speculative" (31, emphasis added). While Jockers is careful to contextualize his own engagement in macroanalysis as a search for new perspectives on well-trodden areas of literary influence and authorial agency,

[37] Jennifer Helene Maher, Helen J. Burgess, and Tim Menzies compare this aspiration towards a metahuman "numerical certitude" via data science to Liebniz's notions of a thought calculator and universal character. Contra McAfee's claim for computational certitude, they argue that big data "propagate[s] rather than eliminate[s] rhetoric" (190-91).

[38] Adeline Koh also makes the argument that "[t]oo many" digital humanists "prize method without excavating the theoretical underpinnings and social consequences of method" ("Letter" 41). Her larger critique pushes back against the notion of digital humanities as a tactical enterprise (cf. Ridolfo and Hart-Davidson as well as Kirschenbaum) that will improve the institutional standing of humanities departments, scholarship, and teaching.

he draws from the understanding that methods are more persuasive when they are "primarily quantitative, primarily empirical, and almost entirely dependent upon computation" (32).

On the one hand it seems reasonable—perhaps even urgent in a post-truth and fake-news culture—to advocate for empirically valid grounds for our arguments, but Cottom speculates that the drive to "[d]atatiz[e]" cultural artifacts "at large scale becomes meaningful not because of its ontological superiority per se but because it rationalizes the hegemonic cultural imperative that all things (and beings) be datatized" ("More Scale"). For Cottom, every computational analysis must theorize power relations and social constructions, and other researchers have also called for us to "unpack goodness in the context of big data" (Maher, Burgess, & Menzies 191). This sort of critical perspective, unsurprisingly, often runs counter to the goals of commercial and political entities involved in the IT sector. A recent and high-profile example involves Google's sudden termination of Timit Gebru, a co-lead for the AI ethics division; the details are expectedly murky but the centerpiece of the controversy is a paper coauthored by Gebru that warns against using large-scale datasets for natural language processing, which is integral to Google's operations and currently an arms race of sorts among AI research and development. Beyond the environmental and financial costs of the required computing power, the authors explain, language models learn from indiscriminate swaths of language available on the world wide web and therefore also learn the biases inherent in that language. At one level, as most web users have experienced, a good amount of language online trades in hate speech, stereotypes, and disingenuous trolling. We also know that what is available online is at best partially representative of the human experience and at worst outright exclusionary, from colonially inflected overrepresentations of Anglophone and Western European languages and cultures to reifications of hegemonic identities along axes such

as class, race, gender, and sexuality. The list continues with problems such as determining the currency of language and the views it expresses (e.g., as they reside in digitized texts), as well as interrogating even well-intentioned language that more commonly associates negative-sentiment words with concepts such as mental illness. Gebru and her coauthors suggest that the industry "invest significant resources into curating and documenting LM training data" and take its cues from "archival history data collection methods" to work towards "a more justice-oriented data collection methodology" (Bender & Gebru, et. al. 6).

Digital humanities scholars have similarly critiqued the "macroscope" as an erosion of the political agency usually exerted in the "close and narrow reading of human experience" (Hitchcock). They describe this political impoverishment as the "ideological purification of data" along with the "ideological suppression of symbol and signification" (Berry, et. al.). In response, they call for what David Berry evokes as a "critical digital humanities," which foregrounds "data-intensive critique" that is "attentive to questions of power, domination, myth and exploitation."[39] In this sense, propositions such as Hallnäs and Redström's slow technology stand as a bulwark against an ethical crisis that seems to affirm one of Michel Foucault's central theses: we all will be rendered into data, and, as such, become objects for the exercise of power.[40]

[39] For a full exploration of the possibilities of a *critical* digital humanities, see the final chapter of David Berry and Anders Fagerjord's *Digital Humanities: Knowledge and Critique in a Digital Age*, Polity Press, 2017. James Dobson takes up this inquiry in *Critical Digital Humanities: The Search for a Methodology*, Univ. of Illinois Press, 2019.

[40] Shoshanna Zuboff updates Foucault's thesis for the digital age with her concept of surveillance capitalism, which "renders all people, things, and processes as computational objects" (i.e., manipulable, atomized data) for analysis and leverage in pursuit of profit and market influence (399).

For teacher-scholars in digital humanities, rhetoric and writing, and other disciplines, a vernacular theory of code provides ways of framing pedagogical approaches that seek to make visible and envision otherwise the design and operations of digital technologies, a move that Steve Holmes, citing Thomas Rickert, describes as "the ongoing unconcealment of the world that functions as the 'background of intelligibility and practical coping from which we work.' " To take a step back, though, the larger question with which this project begins—*what is the value of code for humanities education*?—ought to be interrogated a bit further, given that the notion of value itself must be disambiguated from the speed, efficiency, invisibility, and usability that we might otherwise infer. Especially in a post-COVID world, the realities of ed-tech's rapid growth (1) as the basis for institutional and learning infrastructures; (2) as a business sector influencing higher education's management, curricula, and pedagogy; and (3) as a focus for new credentials and programs of study in technical, professional, and academic areas renders even more urgent the need for a clear understanding of *value* for the use and study of digital technologies in the pedagogical context of the humanities. Along similar lines in the social sciences, Cottom has observed that a "sociology of edtech" is beginning to emerge from a narrower view of how different platforms and applications can solve particular problems such as tracking engagement and ensuring academic integrity. The interdisciplinary inquiry to which Cottom points would consider how technologies of teaching and learning can prompt us to reimagine the goals and structures of education to "further a greater good" ("Rethinking the Context" 29). Whether in the humanities or social sciences, a holistic treatment of *technologies* as a way to assess, critique, and imagine how institutions operate on individuals resonates with Foucault's use of the term "to refer not to tools, machines, or the application of science to industrial production, but rather to

methods and procedures for governing human beings" (Behrent 55). Thinking in this way, our understanding of *value* begins to align with a slow technology framework.

To see the code is to work slowly, to explore without the market pressures of efficiency, to make explicit what is left implicit, and to approach usability in ways that invite participation, dialogue, dissent, and the remaking of the technological medium for purposes beyond what its designers conceived. The prospect of bending platforms inspired my own efforts to use a textual encoding application as the primary composing space for a section of first-year writing in The Ohio State University's Department of English. For this course students wrote and received feedback on all assignments—from daily activities to major essays—on a single, shared XML file using <oXygen/>, an XML editor with built-in validation for the Text Encoding Initiative guidelines. The students' output was updated on a private website that allowed the class to view their work in dynamic ways: for example, tracing the evolution of a thesis statement across drafts, connecting sources to research questions, viewing coded elements alongside those for the rest of the class, foregrounding rhetorical and structural moves, and exploring the work of their peers. The critical response to this course (via an early article published in the *Journal of Digital Humanities*) revealed a potential for more open and participatory pedagogy, especially with the use of markup, which I incorporated into the design and implementation of a junior seminar on archival work, text encoding, and electronic editions in the University of Kentucky's Lewis Honors College. For this course students worked as a research team to examine a collection of correspondence, curate a selection for a digital documentary edition, theorize the role of the edition as well as their roles as editors, determine the best use of TEI guidelines, apply them to the selected texts, and conduct historical research for editorial annotations, interventions, and

framing. As the culminating assessment following an open, constructivist, and participatory pedagogical design, the students and I collaboratively composed a final paper reflecting on their learning experiences and our goals for the digital edition. For a more authentic assessment by way of a real audience and venue, we wrote the essay for the *Midwest Archives Conference Newsletter*. Both courses—the markup-based first-year writing course and the honors seminar on coding and the archives—will serve as case studies in chapters three and four.

As a pivot to the second chapter, I move from establishing a vernacular theory of code to historicizing and surveying the debate over code among digital humanities scholars. The value of code remains as contested for digital humanists as it does among the larger academic community and in the popular imagination. Moreover, digital humanists have long debated the value of code as a medium for the *production* of scholarship in addition to offering textual objects for analysis. Manifestos, self-definitions, and debates over what is and isn't 'DH' have always been part of the scholarly discourse, but code provided the occasion for a lasting provocation during the 2011 Modern Language Association annual conference. This fell during a time that scholars soon described as the "digital humanities moment," when the formerly niche field of offbeat research was ostensibly going mainstream as one of the few boom areas in what otherwise appeared as a bleak humanities job market reeling from the Great Recession of 2008. Perhaps attempting to harness this momentum, digital humanities scholars and organizations often used the "big tent" metaphor to suggest a considerable leeway for what counts as digital humanities work, while deploying *digital humanities* itself as "a tactical term" that would ostensibly grant more traction for research projects, hires, funding opportunities, and organizational changes (Kirschenbaum, "Digital Humanities As/Is a Tactical Term").

Against this rhetoric of expansiveness (or, to put it less charitably, 'anything goes') and market currency, a prominent digital humanities scholar suggested during the MLA conference that coding—i.e., the ability to code, the practice of coding, or leveraging code as integral to one's scholarly production—ought to be the primary criterion for what makes one a digital humanist. Beyond provincial infighting over terms and categories, the debate catalyzed a larger discussion about the value and nature of scholarly work in the humanities. Two discourses develop from this digital humanities moment regarding (1) creative methods and frameworks for scholarly production, drawing from existing notions of critical making or building, and (2) hermeneutical and rhetorical methods and frameworks for analyzing code.[41] Ultimately, these discourses set the stage for writing with code in the first-year composition class and leveraging code in the production of digital scholarly editions, both of which provide a rich testing ground for the imperative to reckon with code in (digital) humanities learning environments.

[41] "Creative" in this sense implicates the notion of *creating*, i.e., making, building, or producing, as opposed to the more notion of novelty or originality.

Chapter 2: Coding (and) the "Digital Humanities Moment"

<Introduction/>

This chapter traces the humanistic debate on coding as it emerged from an academy still reeling from the 2008 Great Recession, when a previously niche digital humanities suddenly found itself launched into the academic mainstream. Matthew K. Gold anoints the early 2010s as "the digital humanities moment" in the inaugural edition of *Debates in the Digital Humanities* and now, roughly a decade following this pronouncement, kairotic appeals that continue to color the discourse on digital humanities in general and coding in particular. In this chapter I begin by unpacking the implications of a provocation during the 2011 Modern Language Association national conference: that to 'count' as a digital humanist, one must know how to code. The reactions and counterarguments drew in part from familiar concerns about the purpose and relevance of humanities scholarship in an academic milieu increasingly ailed by an existential sense of precarity. Beyond these echoes of a well-established overture, the debate also raised new questions about what code can and should be doing for humanities scholarship, how code allows us to think about issues both perennial and unprecedented, and what the evolving nature of code signals for the future of the digital humanities community.

Key to code's standing as a scholarly and pedagogical pursuit in the digital humanities is an ethos of building or making, which in turn reveals contested propositions for what counts as scholarly work and to what degree the pedagogical outcomes of the digital humanities can and should align with competing educational, cultural, and economic value propositions. As an

alternative to the discursive ideation and textual expression that still represents scholarly prestige in the humanities, building aligns with other arguments in areas such as digital and multimodal composition for a more inclusive recognition of knowledge-making. Yet, building also presents its own set of questions for scholars to address. How do humanities scholars acquire the skills to build digital objects with intellectual and technical sophistication? What kind of building counts as a lasting scholarly endeavor, and what is more of an ephemeral exercise? How do we protect theories of building from the productivity imperatives that continue to encroach upon academic labor and culture? And, most importantly, how might an ethics of building address the subjects of our studies as well as the audience who will encounter and use the digital artifacts we make?

Most recently, the emerging fields of critical and rhetorical code studies bridge arguments for building and making with ways of reading, interpreting, and manipulating code as a unique kind of text. Beyond the technical literacies of understanding how code in a particular language can execute successfully using particular platforms, critical and rhetorical approaches look to the perspectives and value propositions evident in code-texts themselves as well as in the rhetorical ecologies in which code-texts are written, rewritten, and circulated. At stake in setting code at the heart of the digital humanities moment is, precisely, to code the digital humanities moment itself. Now more of a historical reference than a contemporary projection, this 'moment' has served as a prompt to reckon with technologies, especially code, in ways that draw from the humanities tradition while adding to its methodological toolbox. In that spirit, the chapter concludes with the opportunity to expand on critical and rhetorical code studies with the use of code in the context of composition and writing studies. This paves the way for the chapter three,

which details the pedagogical case study of a first-year writing course that used extensible markup language as the exclusive medium of composition for all student work.

<"Lines in the Sand": Who Gets to Be a Digital Humanist?/>

The 2011 Modern Language Association (MLA) national conference was a watershed moment for the digital humanities. The discipline, whose practitioners so often weaponize a perceived liminality for claims of avant-garde legitimacy,[42] was the explicit or implicit focus of at least 44 panels during the conference (Sample, "2011 MLA"). The larger irony was not lost in Matthew K. Gold's groundbreaking 2012 collection *Debates in the Digital Humanities*, which in its opening paragraph notes the dramatic progression of the digital humanities from "the next big thing" to "the Thing" over the two or three years leading up to the conference (Gold). Moreover, the social media engagement with questions and topics related to the digital humanities during the conference was remarkably rich at a time when live-tweeting and blogging an academic event was still a relatively new practice. While the buzz, publicity, and hot takes concerning *the digital* as a general phenomenon/concern in humanities scholarship are precisely what Gold has in mind when he frames 2011-12 as "the digital humanities moment," a large part of the debate centered on the controversial proposition of code and coding as the *sine qua non* of scholarly production in the digital humanities. To be a digital humanist or do digital humanities, the claim goes, one must know how to code.

[42] Julianne Nyhan and Andrew Flinn, for example, find a common motif of "underdogs and revolutionaries" in their oral history of humanities computing and digital humanities (270).

Apropos of Gold's unintentionally Carpenterian allusion to digital humanities as "the Thing," the provocative clarity of exactly who counts as a digital humanist shook a discipline with an epistemological and methodological ambiguity that otherwise continues to inspire a tangle of questions concerning who really is a practitioner, and whether we are one ourselves. Even to call digital humanities a discipline sidesteps the categorical ambiguity of how it fits among and apart from other fields of study. The *Companion to the Digital Humanities* (2004) opens proudly with the announcement that digital humanities has emerged "as a discipline in its own right," and the founding of flagship organizations and publications such as the Alliance of Digital Humanities Organizations and the *Digital Humanities Quarterly* bolster the appearance of institutionalized disciplinarity (Schreibman, Siemens, and Unsworth). Yet, the zeitgeist claims did not necessarily play out in the material contexts of the academic job market. In the wake of the digital humanities moment, Roopika Risam writes that "[r]umors of a DH takeover are greatly exaggerated" if one looks to the Modern Language Association's *Job Information List* ("DH Jobs"). Between 2013 and 2016, references to digital technologies in junior rhetoric and composition job postings fell by half, but this may indicate "an assumption for digital fluency and pedagogy from applicants in this field" just as well as it may portend a rapid withdrawal of faculty lines (McClain & Murray 10).

The *New Companion to the Digital Humanities* (2016) questions its predecessor's use of institutional and organizational language, e.g., "discipline," and prefers terms such as "methods" and "field of endeavor" to designate the digitally driven work that it surveys (Schreibman, Siemens, & Unsworth xvii-xviii). Much earlier, the question of whether humanities computing constitutes an academic discipline was the topic of a 1999-2000 seminar series held by the

University of Virginia's Institute for Advanced Technology in the Humanities (IATH). Others sidestep the discipline question by theorizing that the digital humanities are part of "a wider 'computational turn' affecting all major disciplines" from chemistry and biology to linguistics and literature (Tenen). However, this excludes areas under the big tent of the digital humanities that do not depend on computational analysis, such as cyberculture studies, interface studies, new media studies, and digital rhetoric and writing. The controversy over coding serves as an illustrative microcosm of these larger debates on the digital humanities, with its most succinct articulation during the 2011 MLA conference. Grounding the more abstract exercises in field definition with a useful case study, the coding question draws from arguments about disciplinary boundaries, knowledge production, labor and expertise, and the purpose of higher education.

Designed as a lightning round featuring a who's who of digital humanists, session 309, titled "The History and Future of Digital Humanities," asked scholars to present brief position papers on a topic within the scope of the session's title. When it came time for Stephen Ramsay to speak, he delivered a statement (which he later characterized as "pithy and underdeveloped") on the topic of "Who's In and Who's Out": a calculated antithesis to the "Big Tent" theme of the conference of the Alliance of Digital Humanities Organizations (otherwise known as the "Digital Humanities Conference") to be held at Stanford University later that year.[43] Asking if "you have to know how to code" to be a digital humanist, Ramsay declared, "I'm a tenured professor of digital humanities and I say 'yes.' " ("Who's In").[44] While his subsequent comments folded

[43] Further adding to the "anything goes" rhetoric is the conference's website banner, featuring a tie-dye image and lettering with floral patterns that evoke American 1960s counterculture and specifically the ethos of radical inclusion and permissibility among hippie subcultures ("Digital Humanities 2011").

[44] Ramsay's comments—both his MLA position paper and his follow-up essay "On Building"—have since been removed from the web, but still can be found using the Internet Archive's WayBack Machine.

coding into the larger framework of building as knowledge production, Ramsay's "lines in the sand"—the lines of disciplines, of his speech, and of code itself—became, in marketing terms, the stickiest sound bite of his three-minute soapbox (Rogers). It's important to note that Ramsay was neither the first nor the last to suggest such a threshold. David Reider, for example, offered a similar provocation a year later during the 2012 Computer and Writing Conference. Contending that "the new basis of writing is algorithmic," not representational, he argued that "if you can't write code, if you can't think with code, if you can't write algorithmically, you may eventually find yourself stuck in the logocentric sands of the past." The pedestal of the MLA conference and the then-novel fishbowl effect of social media's public backchannels amplified Ramsey's comments and called attention to the digital humanities as a matter of public speculation.

For many attendees of the conference and for many more following the subsequent tweets and blogs, Ramsay became an iconoclast for his apparent rejection of the epistemological and methodological inclusivity of the digital humanities. In one HASTAC (Humanities, Arts, Science and Technology Alliance and Collaboratory) blog entry, Ramsay is taken to task for his "essentialist view" that "polarize[s] and de-historize[s]" digital humanities work in favor of a "theorist versus builder" antagonism (Pettiway). On the public-facing website for Brian Croxall's 2011 introduction to digital humanities course at Emory University, a student speculated that if Ramsay's standard were applied, at least in spirit, to other disciplines, we would be left with few true practitioners. In the nineteenth century, the student wonders, were the only "true" biologists the ones who not only used microscopes but also built them? (Marcinowski).

In reply, Ramsay explained that the development of laboratory science in the nineteenth century shows us how "conducting new research very often means inventing the tools that allow

you take accurate measurements": the very problem that digital humanists now confront with "XML, GIS, and things of this sort" ("Response"). In 2012's *Debates in the Digital Humanities,* Ramsay and Geoffrey Rockwell elaborate on a theory of building that, Ramsay asserted during the post-conference brouhaha, underwrote his use of coding as the threshold of "real" digital humanities work. To understand the significance of building in the digital humanities, Ramsay and Rockwell write, we must return to "humanities computing," which fell out of use as a disciplinary term and was replaced with "digital humanities," but which more usefully designates a type of academic labor that stands distinctly apart from "game studies, media studies, cyberculture, edutech," and other pursuits under the putatively big tent of the digital humanities.

At stake for Ramsay and Rockwell is not just whether code, software, tools, and other platforms are taken seriously as intellectually demanding (i.e., difficult) pursuits, but that they are understood as inherently theoretical and hermeneutical on an equal footing with textual forms of humanities scholarship unique to the printed monograph and journal article. Two years before the 2011 MLA conference, Patrik Svensson surveyed the shift in nomenclature from "humanities computing" to "digital humanities," claiming that practitioners of the former maintain "a very instrumental approach to technology," which "has this basic and epistemically grounded role as a tool." Svensson finds that humanities computing primarily designates a shared methodology and object of study (i.e., the application of computational analysis on textual artifacts), while he brackets as "relatively marginal" the theoretical and hermeneutical gains that Ramsay and Rockwell conversely champion. The latter's writing, then, is also an act of recovery; the work ascribed to the traditional origins of the digital humanities must be theorized and interpreted

more rigorously so that, in turn, we may see it as theoretically and hermeneutically rigorous in its own right.

<Coding, Building, and Scholarly Production in the Humanities/>

In the name of legitimacy, Ramsay and Rockwell subject coding and writing to the same inquiry in an attempt to dethrone the latter as the queen of proofs for scholarly activity and value. If we wish to understand what coding is and what it does in the context of humanities scholarship, we also must make explicit the otherwise implicit assumptions about writing. "To ask whether coding is a scholarly act is like asking whether writing is a scholarly act," Ramsay and Rockwell declare. They proceed to align writing with coding as a kind of "technology—or better, the methodology" that a scholar uses to make an "intervention" in a discourse, a discipline, and/or the world. The word *rhetoric* appears only twice in their chapter, but they insist on a rhetorical criterion for scholarly activity; whether writing or coding, speaking or building, the value of scholarly production lies in the agency that it exerts among material and discursive networks and the degree to which it surveys, critiques, clarifies, revises, and proposes theories and interpretations in areas of scholarly concern. Our invocation of particular actions, media, and tools to describe this process, they point out, is purely mentonymical for the "quality of the intervention."

While a contemporary critique of Ramsay and Rockwell suggests that they "strictly demarcate" the construction of knowledge via writing from the construction of knowledge via building (Cecire), I find that even though they insist on the "untranslatability" of knowledge-qua-

building they do not preclude built things from participating in and affecting discourse. The notion of an 'untranslatable' quality to building and built things leads us to consider the narrow range of forms and genres that have been codified within academic discourses as proof of scholarly activity and professional currency for recognition and reward. As Ramsay and Rockwell write, we might assign a written interpretive essay not just so that the codified form and genre allows students to prove that they understand a work of literature, but, more importantly, so that through the act of writing the student is able to discover their understanding: so that "thinking [may] occur in the first place." In other words, the interpretive analysis essay matters less as a standard for assessment and more as an opportunity for experiment, exploration, and discovery. The example recalls Jerome McGann's famous dictum that "to make anything is also to make a speculative foray into a concealed but wished for unknown" (15). To code, then, is not to codify, at least in Ramsay's understanding, and the ethos of building aligns more with experiment and inquiry than it does with compliance—e.g., ensuring that something, like a building, is 'up to code.'

Invoking McGann in an October 2011 talk for the University of Nebraska's Center for Digital Research in the Humanities, William Thomas argues that the problem with building is precisely one of form and genre. Because the "object[s]" of scholarly production in the digital humanities (e.g., an archive, a tool, an algorithm, a dataset, a visualization) do not align with the forms familiar to humanities disciplines (e.g., prospectus, grant, journal article, monograph), they reveal what is otherwise concealed: "[t]he distance between our wish and our object" and the "profoundly unstable and speculative" nature not only of scholarly production in digital media, but also, we may infer, of humanistic inquiry itself. Moreover, the problem of form and genre

lies in the seductive clarity of their lines in the sand (i.e., what scholarship looks like and what it *is* as a formal and generic definition) for professional assessment as opposed to the murkier prospect of evaluating what scholarship *does* as an intervention. In other words, the lines in the sand might as well be lines in the CV (the "object[s]," in Thomas's thinking), while the impact of—the more distant *wish* for—our scholarly work lies in the more ambiguous territory of asserting agency via speculation, inquiry, and, in the language of the mission statement of the University of Nebraska English department: "imaginative reasoning."

Yet, Ramsay and Rockwell remind us, scholars of humanities computing for years have made a case for coding and building as fundamentally interpretive and transformative pursuits. This often goes by the name of *modeling*. In his touchstone *Humanities Computing*, Willard McCarty takes as a starting point "the fundamental dependence of any computing system on an explicit, delimited conception of the world or 'model' of it" (21). McCarty earlier defines the model-concept for humanities computing as "a manipulable knowledge representation" that requires the selection and exclusion of data, as well as an understanding of how material and phenomena may be rendered as data in the first place, in a "provisional, contingent" process concerned precisely with the model's failings and shortcomings as a tool for discovery ("Humanities Computing" 104-5). For *Humanities Computing*, McCarty defines the model more broadly as "a representation of something for purposes of study, or a design for realizing something new" (24).[45] In both cases, the model's heuristic function suggests a theoretical and interpretive intervention germane to the traditional threshold of humanities scholarship.

[45] Notions of modeling and hermeneutics can be found in earlier publications such as the influential computer science textbook *Structure and Interpretation of Computing Programs* (1st ed. 1985). In the foreword, Alan J. Perlis writes that "[e]very computer program is a model, hatched in the mind, of a real or mental process…We change them as our perception of the model deepens, enlarges, generalizes until the model ultimately attains a metastable

Matthew Kirschenbaum rehashes McCarty's foundational theses in a 2009 *Chronicle of Higher Education* op-ed arguing in favor of teaching computer programming to humanities majors. "Programming is about choices and constraints," he argues, "and about how you choose to model some select slice of the world around you in the formal environment of a computer." Like McCarty, Kirschenbaum expresses interest in modeling precisely for its tension with the complexity of lived experience: how models "draw their strength from selective representation." The pedagogical binaries with which Kirschenbaum makes his case resonate strongly, at least for those of us sympathetic to the embattled liberal arts. Vocationalism and instrumentality fall short where nuance and metacognition evoke affinities with traditional areas of study such as writing, literature, and linguistics. Even Kirschembaum's zoological metaphor for career-oriented programming—"professional code monkeys"—implicitly elevates the *human* in the humanities ("Hello Worlds"). Not only does coding belong in the humanistic disciplines, but it belongs precisely because it lends itself to learning outcomes that we consider integral and unique to a humanities education.

<Building(:) A Place for Code in Humanities Education/>

Kirschenbaum's simian pejorative may strike readers as a snobbish twist of the knife that masks the very real and material stakes of arguments regarding the place of code in humanities curricula and pedagogy. While scholars such as Kirschenbaum celebrate the ubiquity of coding

place within still another model with which we struggle...If art interprets our dreams, the computer executes them in the guise of programs" (Abelson & Sussman).

as a fundamentally humanistic pursuit, other critics elevate coding precisely because it seems to present accessible and sustainable opportunities for vocational training. Put simply, and perhaps crudely, the life of the mind doesn't necessarily put food on the table. *Wired* columnist Clive Thompson summarizes the case for teaching coding as a means to gainful employment in an early 2017 article unambiguously titled "The Next Big Blue-Collar Job is Coding." Against the celebrity savant figures behind startups that become tech giants (e.g., Mark Zuckerberg, Steve Jobs), Thompson describes a "solidly middle-class" worker who might "sling Java-Script for their local bank" as opposed to "craft[ing] wild new algorithms" à la "the 'lone genius' coder."[46] In both scenarios, however, we lose a pedagogical sense of building and modeling; learning how to code either prepares the lucky few for the oracular work of Silicon Valley, or it prepares the hoi polloi for the dependable drudgery of 'blue-collar' subsistence.

Despite the scholarly distaste for appeals to employability, Thompson's argument merits attention for its commitment to a faithful representation of "what most programming work actually is." As we stake out a space for coding in the humanities, then, we would do well to reflect on the outcomes that we imagine for our students, especially as foils for what we perceive to be the pedagogical work of the more technical disciplines in an educational culture that is increasingly attuned to workforce readiness. This final term implies an orientation to learning outcomes related to practices, skills, information, and even dispositions that, we assume, are demanded of students once they are employed in particular positions and sectors. It also implies

[46] Thompson may intend to align coding with blue-collar labor in general, but the language of *slinging* code as opposed to *crafting* algorithms may reify a binary of unskilled and skilled labor among the working class, i.e., slinging burgers (as it is ubiquitously articulated) versus craftsman trades. The implicit deprofessionalization of the code slinger (cf. Kirschenbaum's code monkeys) suggests a need for understanding code work not only as use and application but also as critical analysis, evaluation, and creation: the highest orders of thinking and learning outcomes. Cf. the previous chapter's discussion of labor hierarchies and the visibility of algorithms.

an ideological alignment with neoliberalism in which education functions as a supplier of labor and labor training, reducing the costs and risks of hiring and onboarding. At first glance, the humanities have an uphill battle to fight, as we would sooner unpack and challenge the notion of workforce readiness than simply train a student to be workforce ready. Yet, many have proposed that the humanities in fact *better* prepare students for the workforce with skills that are just as technical. (Here, too, we would unpack how notions of the 'technical' and 'professional' have circulated in ways that elevate particular areas of work and study.) For the humanities' emphasis on so-called 'soft' skills—communication, critical thinking, cultural competencies, information literacies, ethical orientations, etc.—the argument goes, students are better prepared to succeed in the information technology sector even more than students who matriculate in IT programs. We see this kind of claim in provocative headlines such as "That 'Useless' Liberal Arts Degree Has Become Tech's Hottest Ticket"[47] and in success stories such as Daniel Stewart Butterfield, who earned bachelors and masters degrees and philosophy before becoming a self-taught coder and founder of photo sharing platform Flickr and communication app Slack. We see it in studies finding that working engineers spend the majority of their time communicating and collaborating in team environments and place more value on those skills than they did when they were students (Sageev and Romanowski 687, 690). We see it in scientists' calls for a more robust training in communication as part of the culture of science education, better preparing graduates to communicate with lay audiences and frame their work in more nuanced ways (Brownwell, Price, & Steinman E6). And, we see it in extended case studies that find an enhanced self-awareness

[47] George Anders, *Forbes*, 29 July 2015, www.forbes.com/sites/georgeanders/2015/07/29/liberal-arts-degree-tech.

and metacognition in students' writing practices as they switch from humanities majors to STEM fields (Beaufort 107).

Yet, a liberal arts or humanities education remains something of "a black box," in the words of one critic (Carlson). It's tempting to see success stories as generalizable evidence, but, as Scott Carlson writes in the *Chronicle,* these stories more likely are outliers that enjoy the structural benefits of power and privilege: that is, the evidence doesn't mean what we think it means. Echoing Thompson's resistance to fetishizing the tech industry's celebrity savants, Carlson suggests that what matters more than the skills (that *may* be) granted by a liberal arts education is the social and material milieu in which that education occurs. In fact, Carlson suggests, the very skills that, we assume, come from the black box of the liberal arts and humanities may actually draw more from the norms acquired from a socialization among elite circles (e.g., at Cambridge, in Butterfield's case): a sort of selection bias that is discipline-agnostic.

This is what matters for the proposition that coding serves as the threshold of digital humanities teaching and scholarship; in the process of defining what we're teaching and why it belongs in a humanities course, we're tempted to invoke either the oracular or the pragmatic figures— i.e., the maverick CEO or the digital assembly line—that undergird the ethos of neoliberal capitalism: an ideological and material superstructure that the humanities have an ethical imperative to critique, and to teach students to critique. On the other hand, the need for broader, de-centered ways of assessing genres and modes of (digital) scholarly work lead us back to the notion of building or *production,* simultaneously invoking ideation and labor. For its double life in the lexicons of scholarship and economics, production most likely always will

provoke tension between the idealistic and the instrumental. However, the building and modeling theorized by McGann, Ramsay, Kirschenbaum, and a host of others ultimately draws from Aristotelian *inventio:*

> a process that engages a *rhetor* (speaker or writer) in examining alternatives: different ways to begin writing and to explore writing situations; diverse ideas, arguments, appeals, and subject matters for reaching new understandings and/or for developing and supporting judgments, theses and insights; and different ways of framing and verifying these judgments. (Lauer 6-7)

Contra critiques that (digital) humanists are selling the souls of their disciplines by adopting the language of capital, we see building as a heuristic, imaginative enterprise that guards the values of liberal arts education, serves the needs of an ever-changing student population, and challenges scholarly bias against newer genres and modalities of knowledge-making.[48]

Yet, some have worried that the rhetoric of disruption and innovation that we leverage in defense of nontraditional scholarship in fact draws from the very ethos of neoliberal capitalism and, specifically, the information technology sector that so often serves as its avant-garde. In resisting the traditional humanities scholarship upon which digital humanities work ostensibly improves, do we further devalue (and, even worse, do we present an insufficiently complex understanding of) the theories and methodologies that we purport to critique? In other words, we ought to theorize and practice *inventio* via building or production in/as digital humanities in ways that attend carefully to the ideologies with which we may unintentionally align ourselves. As we

[48] For these insights I am indebted to a public exchange over Facebook on 18 June 2018 between Jeff Rice and Hugh Burns in response to Stanley Fish's op-ed "Stop Trying to Sell the Humanities" in *The Chronicle of Higher Education*, 17 June 2018, www.chronicle.com/article/Stop-Trying-to-Sell-the/243643.

have learned from the likes of Althusser and Foucault, we are always contained and constrained by ideology. David Golumbia, for example, urges digital humanists to "see how uncomfortably close the doctrine they advocate is to many of the most extreme ideological attacks on higher education mounted the world over by the political right" ("Death of a Discipline" 170-71). Ideology, in Golumbia's thinking, can operate as an unintended network effect in addition to a willful intervention. Put simply, the impact of our actions by way of the positions and perspectives that they bolster is always at least partially beyond our control. Golumba is particularly concerned with the risk of aligning with perspectives that view the humanities as irrelevant and out of touch. By extension, we may think of the need to self-define (especially in perpetuity, as seems to be the case in the digital humanities) as itself an effect of ideology and power in the form of interpellative responses to stereotype threat (in this case, an anticipated stereotype of scholarly illegitimacy). This feedback loop impoverishes scholarly production in the field, Golumbia suggests, because the persistent urge to self-define (what *is* digital humanities?) precludes more rigorous theoretical and methodological critique. As this relates to pedagogy and curricula, the ontological debates and disciplinary flag-planting make an ineffective and potentially alienating introductory frame for students who have little at stake and few or no reference points in the matter (Cordell). Seeking a coherent understanding of a field of study that is likely new to them, a learner experiences "whiplash" as they navigate contradictory thought pieces written for a specialist or insider audience (Callaway, et. al.). To draw these critiques together, we come to understand that digital humanists may balk at blue-collar paeans to coding, modeling, making, and building, but they also may undermine the work of humanities education in their own apologia for the new.

It is this newness that in part explains the sharp responses to Ramsay's MLA speech and subsequent elaborations on coding and building as integral to digital humanities scholarship.[49] These creative acts occur seemingly *ex nihilo* as an entirely new articulation of knowledge that stands in sharp contrast to the iterative and additive process by which traditional scholarly productions make contributions to extant bodies of knowledge. The trailblazing productions prior to the digital humanities 'moment' bolster this sense of mythic, herculean effort whose futurity remains clouded (e.g, McGann's notion of a "speculative foray into a concealed but wished for unknown"). Put another way, if one must code to count as a digital humanist, what exactly is it that one must code or build? A comprehensive digital archive of an author's or artist's corpus such as the Walt Whitman or William Blake archives?[50] A platform for analyzing and visualizing texts such as Stéfan Sinclair and Geoffrey Rockwell's Voyant Tools?[51] The prospects are daunting, with a threshold of skill, scale, and novelty that place building beyond the reach of most would-be digital humanists. As Ted Underwood explains, digital humanities programs lack the curricular and methodological depth to invite novices into a field that behaves more like a "social network" than an equitably structured educational program. "[I]t's almost naive to discuss 'barriers to entry.' There is no entrance to this field. What we have is more like a door

[49] On the other hand, in a pedagogical context, building may not be such a radically new prospect. Robin Wharton writes that the novelty of building as a scholarly orientation in fact reflects the pedagogical "impoverish[ment]" of post-secondary education compared to K-12, which has long engaged students in "the serious work of play." Among other provocations, Wharton asks: "Why is making K-12 classrooms look more like post-secondary classrooms (often through the implementation of technology) decidedly a good idea, but thinking about how we might make post-secondary classrooms look more like K-12 classrooms (e.g., through learning by doing, integrating play, emphasizing social values like sharing and concern for others, etc.) doesn't seem to have caught on in the same way, especially at the institutional or curricular levels?" ("Building").

[50] https://whitmanarchive.org/ and http://www.blakearchive.org/

[51] https://voyant-tools.org/

painted on the wall," Underwood quips.[52] To be fair, advocates of coding and building sought to make the case for these activities as legitimate forms of knowledge-making in the humanities, especially when compared to discursive and textual ideation. Many sought to establish coding and building as valuable in the first place to make room for later conversation about how to make it approachable for wider range of scholars and practitioners. From this effort, we continue to refine a praxis for building that advances its scholarly legitimacy while making more visible and accessible its entry points, pathways, processes, and impacts.

With the importance of building propelled to the fore of debates in and about the digital humanities, more questions followed: what is the nature of building, and of the builder? what methods and productions does building implicate? how is our understanding of building different from how the concept circulates in discursive and material ecologies outside of the humanities, and perhaps even outside of scholarship and education altogether? A few years after Ramsay's provocation at MLA, Jentery Sayers frames the 2017 collection *Making Things and Drawing Boundaries* in a way that downplays notions of "romantic invention" and "individual genius" in favor of a more "quotidian" sense of building as "maintenance or remaking." He elaborates:

> [This framing] does not sound as epic as innovation, as grand as big data, as impactful as disruption, or as concrete as tool-building, either. However, it may most accurately describe both our everyday engagements with technologies and the physical reality of our materials: *not made from scratch but in media res; not transparent platforms but patchworks of memory and practice*. From third-party dependencies and plug-ins to

[52] The digital humanities are by no means excepted from the systemic problems of bias and exclusion that affect all areas of the academy. Scholars have increasingly called attention to biases of race, gender, ability, geography, class, and culture in digital humanities work.

> emulations and repurposed hardware, starting in the middle is not some abstract idea cooked up by critics and artists on high. Negotiation and maintenance constitute the conceptual matter that makes up making. (emphasis original)

Notions of practice, negotiation, and maintenance resonate with Mar Hicks's assessment of the scapegoating of COBOL in the wake of the Coronavirus Aid, Relief, and Economic Security (CARES) Act, passed on 27 March 2020. Necessary for processing stimulus and unemployment payments en masse, COBOL was thrust into the national spotlight as an outdated, yet ubiquitous language responsible for the widespread errors and delays in CARES Act payments. What's worse, the story went, few programmers were familiar enough with it to address the crumbling digital infrastructure. Many popular versions of the story suggested that a cohort of elderly coders had to be coaxed out of retirement while a younger generation collectively scratched its head. However, Hicks explains that Java, not COBOL, was ultimately responsible for the errors, and that young people as well as old had the expertise to troubleshoot COBOL. The incident did reveal the worsening neglect that COBOL-based systems had endured for decades, which Hicks attributes to an "austerity logic" that prefers disruptive, short-term innovations seeking to "monetiz[e] flaws or gaps" rather than "caring for and fixing the systems we all rely on." As a counter, Hicks advocates for a "labor of care" paradigm implicating a stable professional class (programmers, developers, technicians, and so on), the lasting infrastructures that they maintain, and a democratic notion of the user-publics that those infrastructures prioritize and serve.[53]

[53] The 2019 edition of *Debates in the Digital Humanities* dedicates a cluster of brief chapters to the ethics, theories, and practices of care. In her contribution to that section, Bethany Nowviskie connects the notion of care to a "well-established feminist ethic and praxis" that countered earlier moral philosophy and utilitarianism valuing the "fully developed (implicitly masculine) self['s]...ability to stand apart from and reason outside of familial systems and social bonds." This version of a developed self adumbrates the (usually male) figure of the disruptive entrepreneur and Silicon Valley guru. By contrast, the ethic of care that Nowviskie proposes rests on an "appreciation of context,

Similarly, Sayers seeks to unmoor building from the "evangelist entrepreneurialism" that undergirds popular discourse on maker cultures (e.g., the economic lens of Chris Anderson's *Makers: The New Industrial Revolution*). Conceived as "material experimentation," Sayers's take on building in the digital humanities does not follow the teleology of capitalist production that culminates in the marketable product such as an archive, tool, app, or platform. Rather, Sayers's experimental notion of building values critique and inquiry, alternate narratives, fragmentary production, incomplete knowledge, and generative ambiguity. Technologies, code foremost among them, are "not something we control or master but rather a relationship we negotiate with some skepticism and surprise." Sayers seems to take inspiration from Ramsay's hermeneutics of screwing around—a kind of bibliophilic flânerie that values "community, relationship, play," and the "roaming intellect" as a way of making sense of vast archives of digital and digitized texts—as well as from Mark Sample's theory of non-consumptive reading, which proposes the (re)use of computational and empirical analyses as material for "expressive objects" that "[t]urn your data into a story, into a game, into art" ("Poetics").[54]

The concluding chapters of Sayers's collection explore making through a lens of ethics and social justice, refining arguments for play, experimentation, and screwing around with the

interdependence, and vulnerability" and a rejection of the ostensibly objective critical stance in favor of "personal, worldly action and response." Overall, this version of the self is "most complete when in connection with others."

[54] Sample's notion of non-consumptive reading is a play on non-consumptive research, a technical term designating when "a text is not read by a scholar so much as it is processed by a machine." (Humanists may be more familiar with Franco Moretti's similar notion of *distant reading*). As Sample explains, this carries legal implications for the competing interests of mass digitization efforts and copyright holders (e.g., Google Books, HathiTrust). He offers his own *House of Leaves of Grass* (fugitivetexts.net/houseleavesgrass/) as an example of a non-consumptive reading; to create the text, Sample "reassembled both common and unique words and phrases" from Walt Whitman's *Leaves of Grass* (public domain) and Mark Z. Danielewski's *House of Leaves* (copyrighted) to create a "100 trillion stanza-long mashup" whose voluminousness seems to play on both Whitman's tropes of inner and outer multitudes and Danielewski's setting in a house that is infinitely larger on the inside than it is from the outside (embedded within several frame narratives, texts, and perspectives, no less).

reminder that "inclusiveness, transparency, and accountability require deliberate attention to that hermeneutics, its biases, and how making *shapes* our thinking" (Boggs, Reed, & Lindblad, emphasis original). With project examples such as a digital map of a plantation owner's sexual assaults of enslaved women, a conceptual artwork that uses interviews with a convicted murderer sentenced to death, and an audio drama that remixes the recorded speech of the dead with the speech of the living, the authors adumbrate a theory of digital making with inspiration from Levinas's notion of an "ethical invitation to proximity." Notably, the notion of proximity places these theories of building in contrast to distant reading, big data, scaling up, interface mediation, fast technology, and computational impartiality. Builders—or composers, in the language of composition studies—find themselves in "entanglements" with the subjects of their work, with "blurred lines of us, them, me, her, our voices, and their voices." Much of this entanglement results from the time and labor that builders spend with their subject and materials: hours, days, weeks spent synching voice and breath in audio tracks, poring over manuscripts and written records, recounting horrific events in ways that attend to the dignity of victims (Anderson & Campbell).[55] Broadly, digital humanities practitioners have used language such as 'crafty,' 'artisanal,' and 'DIY' to describe the intimate relationship that develops through their methods of study and production. In other words, the ethical is enacted through the practice of what we might call *close building*. The two pedagogical case studies in this dissertation engage students precisely through close building, which departs from the orthodoxies of teaching and scholarship

[55] Similar to this idea of entanglement, Bethany Nowviskie looks to feminist scholar Nel Noddings's notion of engrossment to explain an "ethic and praxis of care" in the digital humanities: a "close attention to and focus on the other that provoke a productive appreciation of the standpoint or position of the cared-for person or group—or...of the qualities and affordances of an artifact, document, collection, or system requiring study or curation."

(i.e., what would be expected as evidence of knowledge-making and learning) with a praxis of entanglement—perhaps even a labor of care—all as a result of 'seeing' the code.

<Critical and Rhetorical Code Studies/>

Matthew K. Gold's *Debates in the Digital Humanities* frames the period of time between 2009 and 2012 as "the digital humanities moment." For Gold, five phenomena merged to create this moment:

1. a swell of institutional support for digital humanities via new hires, centers, initiatives, and grants;
2. a sense that the digital humanities were uniquely positioned to meet the emerging challenges in higher education, e.g., demands for new research and teaching methods, the battle for the relevance of the humanities, shifts in labor practices and pressure on the tenure system;
3. an unprecedented vigor and publicity in debates about the nature of digital humanities as a field (with Ramsay's provocation as the touchstone);
4. a rapid expansion in the ranks of digital humanists, disrupting any "like-minded[ness]" that scholars might have shared previously; and, implicitly,
5. the ongoing project of *Debates in the Digital Humanities* as a manifestation of these phenomena and publication of record for the ephemeral, peripheral, and accelerating developments in the field.

Gold and other scholars would not have expected any strong consensus on the meaning of the digital humanities moment, and subsequent years have seen this ambiguity become a generative force for scholarship and scholarly production. In particular, the digital humanities moment called attention to code as a critical scholarly pursuit for the humanities, serving a dual role as both an object of study *and* a medium for discovery and expression. While Ramsay was clear about coding as the coin of the digital humanities realm, a topic modeling analysis has recently shown that writing that seeks to define the digital humanities "usually br[ings] up *questions* about coding and digital humanities rather than taking a hard stance about coding's role in the field" (Callaway, et. al., emphasis mine).[56] These questions extend far beyond the technical affordances of code; as one example, Miram Posner critiques the "exhort[ations]" to code for failing to account for structural barriers and disincentives that have excluded women and people of color from participating in programming cultures ("Some Things"). Along the same lines, Adeline Koh questions the "social contract" of digital humanities—a contract that requires both "niceness/civility" and "technical knowledge," mainly coding—on the basis of how social contract models privilege dominant groups, perpetuate inequalities, and exclude different forms of knowledge-making ("Niceness" 100-01). In other words, the legacy of the digital humanities moment is not the imperative to code or know how to code in particular ways; it is, rather, the imperative *to reckon with code* in our scholarly and pedagogical endeavors. To recast this

[56] Starting with the "Further Reading" appendix in *Defining Digital Humanities: A Reader* (blogs.ucl.ac.uk/defining dh/further-reading/), Callaway, et. al. collected and analyzed 334 pieces of writing ranging from blog posts to peer-reviewed journal articles. Reflecting on the prospect of coding as a prerequisite for practicing digital humanities, the authors, who identify as "relative newcomers to digital humanities," found that "[t]he topic model itself proved the least interesting and least difficult aspect of [the] work" while "the really interesting stories emerged when [they] combined the topic model outputs with [their] painstakingly-gathered metadata or after hours of focused reading of the individual documents in the corpus." In terms of the technical methods, in fact, the authors found that the most generative parts of the process were the steps "that one could do with a simple spreadsheet and google charts."

imperative as a reckoning (and not as a social contract) invites a polyvocality and intellectual diversity that Posner, Koh, and others find lacking in stricter definitions.

Among a wide range of digital media and modalities, code is uniquely textual. It is composed of written text and other characters, and it often represents or manipulates other written texts or data. It is, in other words, a text *to* read and write and a text *that* reads and writes. In the collaboratively authored *10 PRINT CHR$(205.5+RND(1)); : GOTO 10*, Nick Montfort, et. al. describe code as "a diary from a forgotten past" (3). Poetic at first glance, the comparison intentionally resists the notion of code as a transparent, mathematical purity in favor of a textual opacity familiar to literary and historical scholars. One critic compares the experience of reading an interpretation of code to reading a detective novel, with the "twists and turns in the analysis of the code and surrounding sources" (Ackermans 1012).[57] Hardly self-evident, a code-text's meaning must be pieced together with patience and from many vantage points. *10 PRINT* draws explicitly from the thinking of Mark C. Marino, who in 2006 synthesized work from media theorists and codework artists to establish a new field and method for understanding code as a textual phenomenon:

> I would like to propose that we no longer speak of the code as a text in metaphorical terms, but that we begin to analyze and explicate code as a text, as a sign system with its own rhetoric, as verbal communication that possesses significance in excess of its functional utility. While computer scientists can theorize on the most useful approaches to code, humanities scholars can help by conjecturing on the meaning of code to all those

[57] Ackermans makes the comparison in her review essay for Mark C. Marino's *Critical Code Studies* monograph and companion website. The detective novel reference concerns Marino's reading of the code behind the "Climategate" scandal when portions of code in leaked emails from the University of East Anglia's Climatic Research Unit seemed to suggest that climate scientists were manipulating data to support climate change theory.

> who encounter it both directly by reading it or indirectly by encountering the effects of the programs it creates. In effect, I am proposing that we can read and explicate code the way we might explicate a work of literature in a new field of inquiry that I call Critical Code Studies (CCS).[58]

For Marino, the cusp of 2007 signaled "the moment for Critical Code Studies." Whether one wraps this into a backdated timeline for Gold's larger digital humanities moment or considers it a prelude on its own terms, exigency itself emerges as a common appeal. Perhaps this is the legacy of the digital humanities moment beyond any particular development: a felt sense that *now* is the time to act. In *Critical Code Studies*, the 2020 monograph that distills and advances insights from roughly fifteen years of development in the field, Marino reaffirms in the present tense that "[i]t is time to develop methods of tracing the meaning of code" (5). The haptic connotations of *tracing* anticipate Marino's sense of critically reading and interpreting code in ways that also resonate with chapter one's vernacular theory of seeing code. Rejecting the notion that reading code is about "the discovery of hidden secrets" (i.e., the kind of deciphering that looks *past* the code or that, in Cypher's language, doesn't even see the code), Marino affirms an approach that attends to "encoded structures, models, and formulations...connotations and denotations of specific coding choices…[aspects of] code's development that are not apparent in the functioning of the software alone," and that treats code as "a unique semiotic form of discourse"

[58] Marino had been workshopping and discussing Critical Code Studies publicly since at least February 2005, with the blog post "Critical Code Studies" on *WRT: Writer Response Theory*, a website run by Marino and colleagues. As Though the post has been removed along with its follow-up, it may be accessed using the WayBack Machine at web.archive.org/web/20061010013523/http://writerresponsetheory.org/wordpress/2005/02/05/critical-code-studies/. As Marino relates in *Critical Code Studies*, he began to conceive of the field in 2004 while studying chatbots (18).

(16-17). It is, in a way, an erotics of code explicitly set apart from more distant approaches to reading in the digital humanities (32).

Marino's approach is deeply rhetorical. With a brief example analyzing portions of the e-voting software Votebox, he outlines the process of reading and interpreting code via the methodologies of critical code studies. First, Marino attends to the historical and cultural context of the problem that the code or software seeks to solve, i.e., the occasion or exigency: in this case the questions of legitimacy that have dogged voting systems particularly since the 2000 U.S. presidential election, that gained new urgency in the wake of the 2016 election, and, in a way that Marino never could have anticipated, that were part of the nation-shaking conspiracy theories following the 2020 election. Marino next considers "the general class of software," i.e., the genre: its conventions, limitations, affordances, and developments over time. Marino here focuses on a tension between proprietary and open-source software as a particularly salient characteristic of the e-voting genre of code. Zeroing in on Votebox itself, he looks to the context and purposes behind its authorship: "who created it, where, when, and why" (23-24). Marino proceeds to examine a passage of the code as well as its documentation, teasing out possible readings and implications for "key constructs of the software" from the assumed masculine gender of the voter evident in documentation to the code's ceaseless vigilance for "challenge events" at any point during the voting process. Marino also considers the coding language itself as well as software and hardware infrastructures on which it relies (26). How do Java and object-oriented programming languages in general figure the voting process, with its key actors and actions? How would other languages imagine and implement the task differently? Scholars in cultural, comparative, and non-English rhetorics similarly examine the unique ways of

communicating and persuading in linguistic contexts with respect to their immanent "locus of logic and relations" (Mao 453).

Critical code studies also attends to cultural contexts; Marino notes a "prevalence of English in high-level programming languages and the basic tokens of programming" that, he argues, often cloaks colonial ideologies in the pragmatism of how coding-qua-English grants access to the majority of libraries, databases, and functions on which programmers need to call (152). Some experimental code projects have in turn begun to explore the potential for non-English-based programming languages. Ramsey Nasser's قلب, for example, is a coding language based in the Arabic writing system; according to his professional website, Nasser has completed "Hello World, Fibonacci, and Conway's Game of Life" in قلب (pronounced *'alb* [ɛlb]). Beyond the strictly linguistic aspects of a non-English writing system, Nasser also suggests that the cultural aspects of language are critical to the code that they produce. Given Arabic cultures' treatment of written script as visual art, for the قلب project calligraphy pieces were produced, "in effect treating the algorithms as high poetry" ("قلب"). Below is a function for calculating the Fibonacci sequence in Python, followed by Figure 8, which shows that same function in قلب. It is worth noting that providing a screenshot of the قلب function was more expedient than typing in machine-readable characters because the Google Docs platform on which this dissertation was initially drafted could not accurately render the arrangement and alignment of the قلب code.

```
def fibRec(n):
  if n < 2:
      return n
  else:
      return fibRec(n-1) + fibRec(n-2)
```

```
(حدد  فيبوناتشي (لامدا  (ن)
(إذا   (أصغر؟ ن ٢)
ن
(جمع (فيبوناتشي (طرح ن ١) )
(فيبوناتشي (طرح ن ٢) ) ) ) ) )
```

Figure 8: the Fibonacci function in قلب (Nasser, "Personal Computer" 28)

For Nasser, قلب is an experiment that highlights the inadequacies and barriers of English as the exclusive source language for coding while also revealing the impossibility of designing a programming language entirely from a non-English source.[59] Such an attempt is "ultimately doomed to failure" because

> [n]on-English programming projects are confronted with an impossible choice: cling to your conceptual and political purity and be cut off from the world of software, or abandon purity, allow English identifiers, and defeat your own purpose for existing… Purely non-English languages could never talk to the web, or email, or any other protocol based on English language. They could not build on the sixty years of software libraries written using English names, and would have to reinvent it all from scratch themselves, siloing them off from the rest of the world and from history, which is both unrealistic and undesirable. ("Personal Computer" 33-34)

Proposing that "the problem is not technical or computer-specific, but cultural and linguistic," Nasser crosses out other possible solutions; for instance, efforts to translate English-based code corpora would never rise beyond experimental, boutique, and small scales given that automated

[59] As Marino recounts, programming languages first operated mathematically. The development of FLOW-MATIC (a precursor to COBOL) in the late 1950s under Grace Hopper represents the first attempt to mimic elements of natural language to make the platform accessible to a larger user base of non-specialists in the private and public sectors. In Hopper's recollection, the more significant barrier was a lack of will to learn the symbols and notations of mathematically oriented code, especially among male-dominated fields of business, politics, and military (153-54).

translation is inaccurate for known language elements and entirely unable to contend with the novel lexicons that programming languages generate, e.g., `def fibRec(n)` in the Python excerpt above. In lieu of offering a concrete alternative, Nasser offers inspiration in the lingua franca that developed among historical trade networks as he wonders how code, too, can unfold organically among global networks of programmers and informational technology professionals (34-35).[60] Though he characterizes this vision as "a fantasy," Nasser nevertheless calls on us to decenter the English language and Euro-American culture in ways that encourage critical and close building through a postcolonial lens. Marino features Nasser's قلب in a shortlist that also includes the ʻAnuʻu project—an effort to translate the C# programming language into ʻōlelo Hawaiʻi—and "Four Generations," a program that uses Cree mathematics to generate beaded portraits.

Reflecting Nasser's argument that these sorts of problems with code have much more to do with culture and language than they have to do with technology and computers, scholars have recently explored code using a rhetorical framework as part of the emerging field of rhetorical code studies.[61] In the introduction to a 2016 issue of *Computational Culture* dedicated to rhetoric

[60] Interestingly, Marino invokes the notion of lingua franca negatively, in terms of the "erosive context" of global Englishes. Even if an anti-colonial programming project succeeds only in replacing English characters and tokens with those of a colonized language, Marino contends, it will have decentered English as the language of digital technology (and thus as the language of innovation and progress) and aligned the code with the pedagogical aims of indigenous language immersion and preservation programs (154-55). Internet linguist Gretchen McCulloch also looks to the past for intimations of the diversification of programming languages; she suggests that English may be to coding what Latin was to writing during the middle ages: a linguistic dominance over communication technology that eventually spread to the vernacular tongues.

[61] Tracing the history of an idea or approach is always incomplete, and the proposed origin point of the 2010s can be misleading. Here I mean to invoke the emergence of an awareness of rhetorical code studies as a concerted field of study or area of inquiry as opposed to the conceptual or methodological origins of the field per se. Many cite Donald Knuth's 1984 *Literate Programming* as a precedent, and Marino devotes a chapter of *Critical Code Studies* to the work of media theorist Friedrich Kittler, particularly with respect to the 1992 essay "There Is No Software." Kevin Brock finds inspiration in Carolyn Miller's 1979 article "A Humanistic Rationale for Technical Writing." Vee and Brown highlight the work of computers and writing scholars theorizing the rhetorical dimensions of computers since the 1980s, and they look even further back to Edmond Callis Berkeley's 1930 commencement address on modern methods of thinking, which places classical rhetoric alongside Boolean logic in the longue durée of "philosophies of

and computation, Annette Vee and James J. Brown, Jr. follow "calls for an understanding of code as an expressive writing practice" and propose that rhetoric provides an opportunity for "interpret[ing] the ways that computation addresses and responds to various audiences and exigencies, makes assertions about identities, and ultimately participates in a complex ecology of forces that shape behavior and perception." Rhetoric studies has historically existed in a state of tension over the appropriate boundaries of field, particularly with claims of purview implicating "any human relation whatsoever" such as Kenneth Burke's capacious frameworks of dramatism and symbolic action. Vee and Brown alternatively conceive of rhetoric as a useful collection of theories and techniques to enrich the study of code and computation as opposed to supplanting it entirely.

Especially salient for the (in)visibility of code, two of the issue's articles engage the work of Carolyn Miller. In her essay "Should We Name the Tools? Concealing and Revealing the Arts of Rhetoric," Miller investigates the longstanding proposition that rhetorical efficacy relies on concealment (or in other words, on not calling attention to the strategies that a rhetor employs) to convey a sense of ostensibly natural and sincere communication. This ethos of authenticity suggests a 'real me' who represents ideas and intentions truthfully as opposed to a 'false me' who engages in a calculated act of persuasion; language in these scenarios is either mimetic or agonistic, the audience naive or cynical. To be believable, Miller memorably writes, "you must conceal your strategies of concealment" (28).[62] Yet, if we are to study these strategies, they must

rigorous reasoning and symbolic logic." Brock similarly suggests that rhetoric has always been algorithmic in its approach to language and persuasion, particularly with respect to the central figure of the enthymeme (43).

[62] Influential in my turn here to Miller are comments made by William C. Kurlinkus re: the necessary dissimulation of rhetoric in response to my presentation "Bathygraphy: The Depths and Surfaces of Electronic Texts" during the Writing Matters in a Changing World Conference at The Ohio State University in Columbus, OH on 2 Feb. 2013.

be revealed, categorized, analyzed, and evaluated; they become disenchanted as the promise of mimesis yields to the instrumentality of agonism. Miller places this tension among the scholastic traditions of *rhetorica docens* and *rhetorica utens*, or rhetorical education and rhetorical practice. As a possible outcome, she postulates an "endless regress" characterized by "a continual escalation of cunning concealment and critical unmasking" (32). Put another way, if rhetorical education names the techniques of persuasion, rhetorical practice becomes all the more artful, requiring more sophisticated education, and so on.

Perhaps ironically, Miller's analysis leaves little room for optimism. Rhetorical education seems to offer scant possibility for the cultivation of rhetors invested in "cooperative" rhetorical practice focused on "the public good" (33). Even if we do encounter the 'good' kind of *rhetorica utens*, we become less receptive and more cynical if we can discern a rhetor's *techne*. Put simply, we're less persuadable when we know we're being persuaded. What does this mean for *rhetorica docens* grounded in the study of code? In the previous chapter I have explored theoretical and orientations to digital humanities pedagogy based on revealing how technologies operate via code. To extend Miller's argument, does seeing the code risk the same endless regress as does seeing the rhetoric? At first glance my claim is seemingly the opposite of Miller's; by unmasking the code, the student-rhetor is better positioned to exert their own agency as a critical user. They become more aware of technologies' influence on their lived experience and are better equipped to think and act beyond the epistemological and rhetorical limitations of any particular platform. Channeling Miller, however, Steve Holmes warns that *merely* seeing the code does not produce enlightened or ethical subject-users, nor can the code be known in its entirety. Like Levinas's notion of the Other, we may become entangled with or in proximity to code, but we will never be

able to unmask its "essence." For Holmes, code is not exactly hidden and not exactly revealed. His case study of how player communities responded to the problem of infinite city sprawl in *FreeCiv* (an open source version of Sid Meyer's popular *Civilization* game series) implicates code's rhizomatic and iterative nature, as well as the unclear boundary between programmer and player in the game's metadiscourse. Mark Marino best demonstrates the significance of Holmes's argument in his reading of the leaked code at the center of the 2009 "Climategate" scandal when internal emails were obtained from the University of East Anglia's Climatic Research Unit. To code-literate readers, the leaked portions could be interpreted as evidence that climate scientists were manipulating data to legitimize climate change theory. As Marino shows, however, readers must also understand the context of how that code functions in a large-scale research project that spans decades. Ultimately, the code initially and somewhat histrionically described as a smoking gun turns out to be something much more banal, like a note scribbled to oneself and pinned on a corkboard with no design to represent the project's overall analyses and findings.

Albeit somewhat inaccessible in its commitments to abstract theory, Holmes's analysis reiterates the importance of close building with a labor of care as the pedagogical framework of *rhetorica docens*.[63] The semantic duality of "crafty" is instructive here, similar to "gaming" the system; we lead students to see the code not so they can command more sophisticated deceptions

[63] Mark Sample's reading of *Micropolis* (an open-source version of *SimCity*) and *JFK: Reloaded* offers another example of engaging code in a way that is amenable to Holmes's critique and defends the value of students getting 'under the hood' with a digital text-object. Examining assumptions how crime works in *Micropolis* and misogynist rhetoric in the paracode of *JFK: Reloaded*, Sample concludes that "procedural literacy must not be strictly limited to reading or writing code, but must also extend outward to language and cultural practices" ("Criminal Code").

—as a hyperbolic example, founding the next Cambridge Analytica[64]—but, instead, so they can explore the creative possibilities of code as terministic screen for human and computer agents.[65] As John Jones and Lavinia Hirsu write in their introduction to *Rhetorical Machines*, "if we want to be part of the future codes of communication, we need to dwell more often in the layers of our surrounding machines, listening to their languages and paying attention to their noise" (8). For information and computer scientists, this means (re)programming pedagogies and curricula so that students are not led "to view programs as powerful rather than perilous, data as abstract and free of bias, and programmers as clever wizards rather than social actors" (Ko, et. al.). Closing out Jones and Hirsu's collection and implicitly crowning it as the successor to her edited issue of *Computational Culture,* Annette Vee notes that rhetorical machines call for "full stack rhetoric," a term borrowed from web development that designates all aspects of the work from the front-end (e.g., user interface of web pages) to the back-end (e.g., server-side maintenance). Yet, Vee acknowledges, this full stack rhetoric does not require the "unicorn" rhetorician who commands a wide array of critical and technical literacies. "[F]ull-stack rhetoric is really only possible as a collaboration," Vee writes, decentering the lone decoder and, in a way, obviating the charge à la Holmes of an all-knowing rhetor and an entirely knowable code ("Full Stack Rhetoric" 241).[66]

---

[64] Now defunct, the political consulting firm Cambridge Analytica became a fixture of the discourse on the digital targeting and advertising tactics of the Trump campaign during the 2016 U.S. presidential election. Much of the debate centered on the question of users' data privacy on Facebook, but another source of controversy was how that data was used to develop and distribute advertisements designed to prey on specific user-audiences' likely biases as determined by algorithmic analyses of any and all information that could be harvested from their Facebook activity.

[65] Given the plurality of meanings for "screen" in digital humanities discourses, I will clarify that I mean here to invoke Kenneth Burke's notion of terministic screens as articulated in *Language as Symbolic Action* (University of California Press, 1966).

[66] I am reminded here of Bethany Monea's *Kairos* webtext "Screen Reading" (discussed in chapter one) wherein we view the layers and infrastructures that a single codetext implicates. Titled "Down the Rabbit Hole," a slideshow moves us from an Amazon product webpage as it would display in a browser, to the markup language that underlies that browser-based GUI, to the circuit boards (and, by implication, the computer language running them) on the

Accordingly, Kevin Brock situates rhetorical code studies among critical code studies, software studies, platform studies, technical communication, and computers and writing as a network of overlapping inquiries that bring particular strengths to the project of understanding code from a digital humanities perspective. Drawing from "rhetoric's tradition of focusing on the means by which rhetorical agents attempt to induce specific audiences to various kinds of action," Brock establishes that the focus of rhetorical code studies is the "goals that developers, software users, or even technological systems work toward and the types of meaning making they engage in in order to achieve those goals" (30). While algorithms are "the concept most central" to Brock's essay on this emergent field, he treats them less as a recipe—as in many definitions that cleave to a notion of steps for solving a problem or performing an operation—and more as an approach to making arguments and persuading audiences (33). The enthymeme, for example, which in Brock's thinking is *the* central concept of Western rhetoric, "functions algorithmically by leaving implicit reference(s) to one or more of the syllogism's components so that an audience will compute the logic of those missing components," thereby making meaning and taking part in the persuasive act themselves (41-2). Brock is not alone in finding ways that rhetorical concepts and methods might illuminate code and algorithmic culture in novel ways. Anthony Stagliano, for example, examines CV Dazzle—an effort to thwart facial recognition technologies—as an exercise in "material *metis*, a wily, cunning response to a hostile situation" (177). Designing makeup, hairstyles, and other cosmetic features, CV Dazzle sought to confound the algorithmic definitions of the human face based on the Viola-Jones Algorithm, the progenitor

user console that run the code, to the server-side storage devices and facilities, to the land and warehouse structures that house the servers.

of current AI facial recognition and analysis. Overall, Brock frames the rhetorical situation as an algorithmic calculus of variables with infinite variation based on the conditions and constraints of the moment. His scope is expansive, ranging from an analysis of discourse *about* code among programmer communities to readings of code itself, most prominently the open-source Mozilla Firefox web browser. Ultimately, Brock urges rhetoricians to look past the user interface that so often represents the terminus of critical analysis and consider the code that generates it. In short, he urges us to *see* code—not as a static, isolated text, but rather as a moment in a larger process of negotiation and persuasion that involves many rhetors, audiences, platforms, and texts.

Following my sense that the mandate of the digital humanities moment is to reckon with code, and following the idea of full-stack rhetoric as a collaborative, interdisciplinary enterprise, the remaining chapters leverage adjacent discourses and case studies as ways of expanding the repertoire of critical and rhetorical code studies. In the next chapter I turn to composition studies as a ground for demonstrating the value of coding in the context of the learning goals for first-year writing. Brock connects the composition of code to the classical rhetorical concept of *progymnasmata*, or imitation exercises, and he examines several case studies in JavaScript (152). He reserves his only explicit references to teaching for these exercises, hoping that they will lay the groundwork for "improving relevant rhetorical and programming pedagogy and further the push for computational and procedural literacy" (184). Building from this precedent, I reflect on my students' use of markup language to complete a semester's worth of coursework in first-year writing. This experiment differs from Brock's examples of composing with code in that the goal was not to imitate or produce functional code objects (e.g., JavaScript modules) but rather to use code as an unfamiliar medium that attuned students to reflect on their choices as writers. This

divergence addresses a gap in critical and rhetorical code studies regarding pedagogy: what value does code bring to a humanities education, especially at the undergraduate level, where the end goal is not necessarily to prepare students for programming or development? To embrace code fully as a means of composition, how does it allow us to invent, reflect, and persuade, and to what degree may we align code-based pedagogy with a range of learning outcomes related to *the critical*? For its grounding in pedagogy, practice, and building ideas and arguments, writing and composition studies lends a unique perspective on code as both a heuristic device and means of expression, all of which complements critical code studies and rhetorical codes studies' focus on communicative and persuasive dimensions. Further, code provides a fresh avenue for writing teacher-scholars to lead students toward learning goals focused on process, reflection, and decision-making, all grounded in a *kairotic* medium that lends a unique sense of urgency and impact to what often is dismissed as a curricular requirement to 'get out of the way.'

Chapter 3: Composing with Markup Language in the First-Year Writing Class

In the age of new media, there is no way to avoid markup.
Markup is text. Markup is communication. Markup is writing.

Bradley Dilger and Jeff Rice, From A to <A>: Keywords of Markup

<Introduction/>

In the previous two chapters I have outlined a vernacular theory of code for humanities pedagogy and traced the discourse on code in the digital humanities from the generative 'digital humanities moment' in the early 2010s to the present.[67] My theoretical argument rests on the notion of 'seeing the code' in a framework informed by critical digital pedagogy and learning outcomes across the humanities disciplines. The value of code for humanities education lies in its ability to make visible the work and propositions of programmed and encoded text-objects in a way that invites learners to contextualize, evaluate, revise, and recreate them for purposes unconstrained by hegemonic and instrumental logics. At the same time, scholars have contested the value of code/coding as a central object of study and scholarly activity in the digital humanities. Earlier provocations regarding whether digital humanists should know how to code have paved the way for a continuing reevaluation of the standards for scholarly production in the

[67] Portions of this chapter have appeared in a different form in the *Journal of Digital Humanities*, vol. 2, no. 2, 2013, under the title "Changing Medium, Transforming Composition," at journalofdigitalhumanities.org/2-2/changing-medium-transforming-composition-by-trey-conatser/. The article was itself a revision of a HASTAC blog post from 18 April 2013 under the same title at www.hastac.org/blogs/conatser4/2013/04/18/changing-medium-transforming-composition. 'The present' for the purposes of this dissertation refers roughly to a time spanning 2020-2022.

humanities especially with respect to work that falls outside the boundaries of print-based discursive ideation and argumentation. One branch of code scholarship has explored the potential for building or making as a legitimate form of scholarly expression, while new fields such as critical code studies and rhetorical code studies have engaged code as an expressive medium that benefits from and expands the hermeneutical methodologies that are essential to humanities teaching and scholarship.

These intertwined lines of inquiry have shown that the mandate of the digital humanities moment is not necessarily to know how to code in a particular way or to a certain proficiency, but to reckon with code as an ambient medium and emergent mass literacy. Scholars such as Mark Marino and Kevin Brock have advanced considerations of how reckoning with code fits with the learning goals and environments of the humanities in the nascent fields of critical code studies and rhetorical code studies. In this chapter I look to composition and writing studies as an adjacent field for building on the work of Marino, Brock, and others in imagining how coding not only fits into but also transforms the nature of teaching, learning, and scholarship in the field. I take as a cue the ubiquitous comparisons between coding and writing that find their most concentrated expression in the work of Annette Vee (most significantly, her monograph *Coding Literacy*). Matthew Kirschenbaum has described code as a medium characterized by "choices and constraints," and these terms surface frequently in the discourse about writing itself, especially when it comes to articulating the intellectual work of writing that transfers as a universal skillset ("Hello Worlds").[68] John Warner, for example, states in the opening of *Why*

[68] Vee also uses this language when writing about the creative uses of code: a "broader phenomenon about the affordances of code, the value of different constraints, and programming's complicated relation to human and machine audiences" (123). In *Toward a Composition Made Whole*, Jody Shipka writes that "what is crucial is that

They Can't Write—a sweeping diagnosis of writing instruction circa 2018 with suggestions for moving towards more meaningful learning experiences—that "[t]o write is make choices, word by word, sentence by sentence, paragraph by paragraph. Writers choose what they want to write about, whom they want to write to, and why they're writing" (5).[69] This notion of reflecting explicitly on one's choices and framing coding and writing as expressions of conceptual and rhetorical models provided the inspiration for the experimental approach to first-year writing that this chapter details as a case study. Overall, this case study demonstrates the value of code for humanities curricula and pedagogy and shows how 'seeing the code' supports and transforms disciplinary work in composition and writing studies.

I draw here from the case-based rationale that Chris Aaron Lindgren offers for his study of coding on a data-journalism team, which in turn draws from Robert K. Yin's *Case Study Research and Applications: Design and Methods* and Anne Haas Dyson and Celia Genishi's *On the Case: Approaches to Language and Literacy Research* (126). Yin likens the "conditions that justify" a single-case study to those that justify a single experimental study; he indicates that a compelling rationale for a single-case design "arises when the case represents an *extreme* or *unusual* case, deviating from theoretical norms or even everyday occurrences" (49-50, emphasis original). Yin also establishes the ground for single-case designs of a "revelatory nature," when

students leave their courses exhibiting a more nuanced awareness of the various choices they make, or even fail to make, throughout the process of producing a text and to carefully consider the effect those choices might have on others" (85). In a case study involving students composing in both text and video to explain a scientific concept, Michael Ennis found that students considered "how they had taken their writing choices for granted, and creating the video prompted them to consider those design choices more carefully" (49).

[69] I first encountered this way of thinking about writing from Edgar Singleton, Director of First-Year Writing in The Ohio State University's Department of English. Training new graduate teaching assistants to be instructors of record for first-year writing, he would often explain the rationale of assignments' parameters and scaffolding as providing "enabling constraints" for students to engage in structured learning experiences that nevertheless kept an intellectual onus for decision-making on the writers themselves.

"a researcher has the opportunity to observe and analyze a phenomenon previously inaccessible to social science inquiry" (50-51). While Yin's foundational text aims to guide empirical research in the social sciences, his rationales for single-case studies engage what Dyson and Genishi write of as "the messy complexity of human experience that leads researchers to case studies in the qualitative or interpretive tradition" (3).[70] The aim of case studies, they explain, is "to see what some phenomenon means as it is socially enacted in a particular case" (10). In this chapter, I ask what code—and markup language in particular—means as it is enacted in the specific academic context of first-year writing courses as well as composition and writing studies at large. It is an unusual case given that the use of markup language significantly deviate from the norms of first-year writing curricula and teaching methods, and it is also revelatory given that no previous effort had sought to implement such a design in/as writing curricula. First-year writing courses in fact offer an abundance of 'messy complexities' that, in this case, yielded rich opportunities for interpreting the work and worth of code in a commonplace curricular experience grounded in the goals and values of a humanities education.

During the 2013 spring semester I taught a section of first-year writing at The Ohio State University for which students composed all assignments, from daily exercises to major essays, in extensible markup language (XML). As an additional constraint, students followed the Text Encoding Initiative (TEI) guidelines: a framework and vocabulary for modeling texts in XML as well as a structure designating where elements can be in relation to others. Assignments required a rhetorically oriented markup scheme attuned to the learning outcomes of both the activity at

[70] Here, Dyson and Genishi cite Frederick Erickson's "Qualitative Methods in Research on Teaching" in the *Handbook of Research on Teaching* 3rd edition, edited by Merlin C. Wittrock, Macmillan, 1986, pp. 119-161.

hand and the course as a whole. Students used the <oXygen/> software, a TEI- based platform that has become a standard choice for digital editing and archival projects in the humanities.[71] Rather than writing and submitting discrete files to the instructor as is the standard practice with learning management systems and word processing programs, students composed all activities and assignments on a single class corpus file. To manage the process of 25 people editing one increasingly lengthy file many times over the course of approximately four months, the class used an online subversion repository: a cloud-based version control system from which students downloaded the current version of the corpus file each time they intended to work on something from the briefest of edits to the most sustained drafting sessions.[72] Once they found themselves at a stopping place they uploaded the revised file, which the subversion repository reconciled with additional edits that other students had uploaded in the interim.

The technical novelty of this approach was minor in comparison with the learning affordances of what I describe as a sustained, radical transparency among students in the class. Students in standard writing courses are occasionally exposed to snapshots and fragments of others' work during peer review activities, but because students in this course always worked on the same document, so to speak, they could see all of their peers' work at all times during all stages of the writing process. At the end of the semester the XML corpus file extended beyond 42000 lines with over 1300 commits, i.e., individual instances of someone submitting an edited version of the corpus file back to the SVN repository. An additional affordance of the corpus file

[71] www.oxygenxml.com

[72] For this class I used Assembla (www.assembla.com/home) primarily because, at the time, it offered a free subversion repository that was not publicly accessible, as was, for example, the now-shuttered Google Code (code.google.com/). Git represents a much more standard method for version control and collaboration, with GitHub as its public-facing 'social media' site for the viewing and exchange of code and projects.

was the ability to transform students' code into an HTML-based website that allowed them to view their own and each other's work in dynamic ways germane to a reflective writing practice, e.g., connecting sources with research questions, viewing the evolution of thesis statements across assignments, filtering for specific types of appeals and other rhetorical moves, comparing isolated passages and whole papers with their peers'. The HTML-based view also allowed students to experience their writing in the more familiar environment of web-based publishing with no visible markup interrupting the reading experience of their written discourse.[73]

Before detailing the course's design, implementation, and reception, I need to establish the theoretical and pedagogical bases that extend the work of this dissertation's first chapter into the areas of writing and composition studies. What, in other words, is the relationship between code and *writing studies*, specifically, as a unique disciplinary environment in the humanities? The fourth chapter engages the more prevalent use of XML for the development of digital scholarly editions through textual editing and encoding, so I do not address those aspects here. Rather, this chapter and the experiment that it frames are concerned mainly with innovative applications of markup language that enhance writing-based learning environments. The use of code in novel contexts offers new possibilities for developing digital literacies in ways that, like writing, lead to discovery, interpretation, and the growth of the critical imagination. As Anette Vee argues, both coding and writing "help to build knowledge" and "function as material

[73] Importantly, students still had access to the XML and HTML markup at the point of the browser-based website, but HTML did not bear as much pedagogical value for the course as I had designed it. While XML is concerned with the *nature* of its content, HTML is concerned with *presentation* or how the content appears on various devices. The distinction between text encoding and presentation is essential for digital scholarly editing and will be taken up in the fourth chapter.

intelligences" in a way that "inextricably intertwines the technological, the social, and the cognitive" (98, 102).

<Coding Literacy and Writing Technologies/>

Writing has long stood as a ready comparison for coding, ranging from the offhanded, expedient analogy to the systematic analysis using frameworks such as speech-act theory. MIT Media Lab faculty and co-creator of the Scratch programming language Mitchel Resnick, for example, has argued that "coding...[i]s an extension of writing," and, like writing, is a heuristic process that enables learning, problem solving, design thinking, creativity, and communication.[74] Understanding code as an *extension* of writing itself and not merely as a similar-but-separate practice, Annette Vee invokes a sociomaterial framework to place code in a continuum with the development of mass print and, before that, alphabetic inscription, all in the category of writing technologies. Vee's monograph *Coding Literacy* is the most sustained analysis to date that takes seriously the implications of the continuing exhortations to code as an emergent literacy of equal consequence as writing. Emphasizing the interrelated social and material dimensions of coding, Vee resists a technological determinism that she finds in earlier media and literacy theories such as Marshall McLuhan's oral and literate man, all of which suggest that technologies themselves (as material presence and process) effect cognitive and social change towards an imagined and ultimately Western hegemonic ideal. In terms of teaching practice, Vee's theoretical orientation

[74] Designed as an educational programming language for children, Scratch coders use a visually driven interface to create games, stories, animations, and other media. See scratch.mit.edu/.

and critique lead us to take a closer look at the benefits and outcomes we ascribe to code-based curricula and pedagogies; in other words, do we assume that teaching coding literacy produces a particular (and particularly desirable) subject? Do we imagine a specific set of outcomes for that subject? What and whom do those outcomes serve? In one possible response to these questions, a code-literate subject acquires more agency to surpass the socioeconomic conditions of their upbringing and assert their own will over structural and systemic forces; yet, this apparent freedom to self-determine is granted within a technocapitalist framework that enumerates the means, ends, value, and raison d'être of labor.

Calls for coding literacy continue to trade in this determinism, which we see in proposals that the acquisition of programming skills offer a path forward for laborers in declining industrial sectors and economic regions. True to Vee's critique, assumptions that learning how to code will transform people's lives often fail to account for sociopolitical dynamics and structural barriers. As one example, the 2017 programming boot camp startup Mined Minds targeted rural workers in West Virginia to great fanfare but quickly became a spectacular failure profiled in the *New York Times*; at the time of reporting the organization had all but imploded and just one graduate had landed a job in programming (Robertson). The sins of Mined Minds extend well beyond a naive understanding of coding literacies, but it is worth noting that in the litany of complaints registered in the *Times* profile, a prominent one was a lack of instructional guidance and meaningful learning activities. The proposition in which Mined Minds traded, it seems, was that exposing learners to coding platforms and procedural instructions for coding—sometimes, apparently, telling students to look for instructions on their own—would catalyze the individual and community prosperity that the organization promised and its students so strongly desired.

We might term this proposal *exposure theory*, the notion that code and coding in their own right cause epistemological, intellectual, and technical evolution. With hindsight and cynical irony, the name Mined Minds somewhat accurately foretold the cycle of pillaging and abandonment that Appalachian communities have endured for generations. As Antonio Byrd argues, the purported social interventions of programming boot camps often rely on "a new coding literacy myth" in the vein of Harvey Graff: namely, that knowing how to read and write code will counter (digital) inequality, provide access to lucrative careers, and "evolve" the technology sector towards more diversity and inclusivity (32). Such a myth implicitly frames problems of access and equity as a supply-side dearth of literate coders and ignores structural barriers that are agnostic to skills and knowledge.[75]

Vee shows that as early as the 1980s researchers were casting doubt on the efficacy of tech- and tool-centered coding instruction as a transformative project *tout court*. She looks to the reception of Logo, a coding language developed by Seymour Papert as part of a modestly successful, boutique effort in British schools of the 1970s that was later implemented in a larger number of classrooms in the United States. Lacking in Papert's theoretical grounding in the work of Jean Piaget and without intentional praxis in day-to-day classroom instruction, the widespread application of Logo suffered from a lack of "good teach[ing] and support to help kids think about procedures and problem solving" (101). A 1984 critique of the U.S. Logo initiative and others like it underscores this very issue. Finding that programming instruction often wrongly assumes that "spontaneous experience with a powerful symbolic system will have beneficial cognitive

[75] Byrd conducted an ethnographic study of low-income, African American adult learners enrolled in a coding boot camp similar to Mined Minds, also in 2017. Their experiences speak to "a complicated system of inherited disadvantages" that demanded creative use of personal resources and support networks to progress through the boot camp's curriculum (34).

consequences, especially for higher order cognitive skills," the authors emphasize that a robust approach to teaching (with) code requires deliberate instructional design, pedagogical praxis, and contextual adaptability (Pea and Kurland 138). Educational coding initiatives have validated these findings; the Story-Writing-Coding platform, for example, was developed in the mid-2010s for British primary students to engage in code-based storytelling and composition in a carefully structured pedagogical framework. The researchers and developers sought to respond to new requirements in England's Primary National Curriculum with a capacious vision for coding "as a form of literacy" that should "broaden [students'] understanding of how meaning is created in multimodal texts (such as code, print text, static and moving images)." Learning activities that involve code should not seek merely to instill programming skills but should aim for learning outcomes that "support [both] conventional and new literacies, especially multimodal literacy" (Price and Price-Mohr 736). A large portion of educational research on coding instruction and curricula focuses on primary and secondary levels, but a literacy framework allows us to speak of code across pedagogical and andragogical contexts as one among many "artistic and powerful forms of communication and creative expression" for which affective, social, and metacognitive aspects of learning are just as important as skill-based masteries (Govind).[76] Attuned to learning needs such as these, often explicitly in its outcomes and curricular design, the writing class offers an opportune space for students to practice coding in ways that bolster literacies new and old. Put

[76] For an earlier overview of this research, see Shuchi Grover and Roy Pea, "Computational Thinking in K-12: A Review of the State of the Field," *Educational Researcher*, vol. 42, no. 1, 2013, pp. 38-43. For an example of more recent applications, see Marina Umaschi Bers, "Coding as Another Language: A Pedagogical Approach for Teaching Computer Science in Early Childhood," *Journal of Computers in Education*, vol. 6, no. 4, 2019, pp. 499-528. Part of Bers's DevTech Research Group at Tufts University, Madhu Govind (quoted here in the main text) reflects on his experiences taking a college-level Python course in relation to his research observations with early childhood learners using block-based platforms such as ScratchJr and KIBO.

more strongly, the pedagogical frameworks of writing and composition offer more possibilities for engaging with code in both imaginative and practical ways.

Taking a sociomaterial approach to writing technologies and literacies, Vee addresses coding with the nuance it requires; neither entirely material nor entirely social, all "literacies circulate in technologies...in symbolic systems...and in networks" (104). Her understanding of how code works as a literacy significantly informs my thinking on its applications in writing and composition curricula, and is worth quoting at length here:

> So, when I say that programming builds new knowledge, I do not mean that the technology of programming inevitably leads to certain cognitive or societal effects. As a technologically mediated symbolic system, programming makes some kinds of thinking more available than did previous technologies of communication. Through the ways it enables people to structure and express information, it uncovers certain kinds of tacit knowledge. Here, I follow Heidegger in his assertion that technologies can *reveal* certain things latent in their design and implementation...Programming and writing are both socially shaped and shaping technologies that have 'become in a very real sense part of our thinking, remembering, and communicating.' " (105, emphasis original)[77]

To consider technologies—and writing technologies such as code in particular—in terms of their revelatory capacity resonates with the theoretical framing in the first chapter of this dissertation, as well as with the sense of writing as a heuristic enterprise which so ubiquitously informs how students practice composition, particularly in first- and second-year courses. Making visible the

[77] The quoted passage with which Vee concludes this section comes from Andrea diSessa's *Changing Minds: Computers, Learning and Literacy*, MIT Press, 2000, p. 6.

models, procedures, arguments, assumptions, and goals that underlie it, code invites critique and imaginative thinking towards what Julia Flanders has described as "building otherwise."[78] Vee's understanding of coding and writing as both "represent[ing] as well as construct[ing] the world" reflects this tension in any utterance or communicative act. Drawing from the speech-act theory of J.L. Austin, Vee considers code as descriptive and performative along three axes: locutionary representation, illocutionary intent, and perlocutionary effect. Following later emendations of Austin's framework, Vee also calls attention to "context, culture, and participants" as critical to interpreting the full import of speech acts beyond the abstract rules that govern them (114-15).[79] As a writing technology, then, code both represents and proposes while coder-writers navigate a spectrum of literal meaning (the code they write), authorial intent (what they want the code to do), and audience response (what the computer does with the code, what the code does or causes to be done to others, and what others do with the code), all affected by contextual influences in social, economic, infrastructural, professional, and political ecologies.

Following a sociomaterial approach and attending to the unique affordances of code as a symbolic and technological system, Vee considers how composing with code is fundamentally different from traditional writing scenarios such as composing with a word processor, on a blog editor, in a messaging app, and, of course, with pen and paper. There is a rich body of literature

[78] In Flanders's extended explanation, an approach that 'builds otherwise' examines "the full stack of technologies" at hand, in particular for how they reify power asymmetries, and seeks more desirable alternatives.

[79] See J.L. Austin, *How to Do Things With Words*, Harvard University Press, 1962; and John R. Searle, *Speech Acts: An Essay in the Philosophy of Language*, Cambridge University Press, 1969. Locutionary, illocutionary, and perlocutionary acts can be distinguished, respectively, as the literal meaning of what was said, the purpose or desire behind what was said, and the effect that the speech achieves. As an example, a host might say to their last remaining guest at a dinner party: "well, it's getting late." The locutionary act signifies that the time is late in the evening. The illocutionary act is to convey that the host desires for the guest to leave without offending them. The perlocutionary act might be for the guest to leave graciously; or, missing the mark, the guest might remain.

examining the impact of particular media, materials, and modalities for writing. From the early days of the world wide web, Christina Haas underscores how material developments in writing technologies influence (but do not solely determine) cultural and cognitive changes in writing practice, from classical debates surrounding the merits of orality-versus-writing to the purported revolution of computer-based writing at the advent of the world wide web (5). N. Katherine Hayles later proposes a methodology of media-specific analysis as a salient approach for understanding digital artifacts, especially hypertext, in both material and symbolic terms. Hayles distinguishes her sense of materiality as dynamic rather than static, emerging from "the interplay between the text as a physical artifact, its conceptual content, and the interpretive activities of readers and writers" ("Print is Flat" 72). Electronic hypertext, in Hayles's analysis, bears unique aspects among which are "fragmentation and recombination," "bilingual" channels of code and natural language, and "distributed cognitive environments" that implicate human and machine actors (74, 84). While Hayles seeks to investigate and complicate the distinctions between print and electronic literature, Vee's analysis looks at code's affordances as a literacy both distinct from and intertwined with writing in both print and electronic forms.

In essence, Vee conducts a media-specific analysis of code through the lenses of literacy and speech-act theory. While typical speech-act theory holds that interactions can be infelicitous, i.e., not conforming to the desired perlocutionary effect, the computer has a perfect alignment (except in cases of mechanical error) between the locutionary act—the code that is written—and the perlocutionary effect: how the code is compiled and executed. In the first chapter I write that there is an implicit tension in the notion that nothing goes without saying in code because no text may be exhaustively comprehensive in truth to lived experience. Vee similarly explains that the

task of "describing procedures perfectly" in code often means that while the computer is always doing exactly what it has been told to do via the code itself, the reality of programming is one of infelicitous interactions between the coder's intent and the computer's operations. She describes this phenomenon as the "*technical* perlocutionary affordance" of code, which leaves room for the indeterminacy that in natural language brackets all communication as we know it (116, emphasis original). In fact, digital media theorists have proposed that glitch and error constitute the normal state of computer-mediated (inter)action despite the computer's apparent capacity for the precise execution of commands and procedures in closed systems.[80] At the same time, code does largely rely on "explicit definitions and form[s]" that are enacted in hierarchical and self-referential ways. While Vee acknowledges that the "fuzzy logic" ostensibly unique to natural language and communication may in fact be within the province of machines if we heed researchers working on the bleeding edge of deep learning and artificial intelligence, her focus rests on code of a day-to-day, ubiquitous flavor: the kind practiced by a wide variety of professionals and amateurs in the familiar categories of C, Python, Ruby, Java, HTML, XML, and so on, all of which rely on similar structural and definitional epistemes (117).[81]

Another critical aspect of code for Vee's analysis is the social context of its composition, circulation, professional practices, and support networks. While Vee does not seek to impose an oversimplified distinction between coding and writing, the apparent differences that she reviews may in fact resonate as similarities when considered from a different perspective. This is, in part,

[80] See, for example, Peter Krapp's *Noise Channels: Glitch and Error in Digital Culture*, University of Minnesota Press, 2011.

[81] Regarding day-to-day, ubiquitous coding, the International Data Corporation found in its worldwide developer census that "[t]he overall developer population in 2020 was 26.2 million and features 13.5 million full-time developers, 7.8 million part-time developers, and 4.9 million non-compensated developers."

an illustration of the nuance that writing and composition studies bring to Vee's literacy-driven framework. In professional programming, for instance, authorship is collaborative, distributed, and fragmented along axes of time, organizational hierarchy, and project management. A coder working for an organization as large as, say, the Online Computer Library Center (OCLC) does not inhabit an authorial role as it is traditionally practiced for writing-based learning activities in humanities coursework, for which writers usually exert sole or significant agency on the totality of the text. Rather, our hypothetical OCLC developer most likely works within the confines of particular tasks or microsites within an overall code-text that does not have clear boundaries and does not exist in a single unified form. They may be tasked with reviewing and revising code that has been reviewed and revised by many others. They may focus on very specific aspects while ignoring other parts of the code. Their contributions are checked, revised, versioned, and merged with many others in a complex digital ecology.[82] Large commercial efforts such as these are made up of "a palimpsest of code and coding practices," Vee writes, while also pointing to the velocity with which code is shared, copied, referenced, and modified on open and closed fora where the traditional rules of authorial attribution need not apply (129).

Yet, we know that writing plays out similarly in professional and political settings with a complex organizational structure. Consider the Affordable Care Act, a.k.a. Obamacare, signed into law on 23 March 2010. How many drafts and versions of the text were managed and merged into who became the official document? How many people—researchers, lawyers, healthcare professionals, staffers, outside consultants, lobbyists, political allies and opponents, etc.—had a

[82] The example is more deliberate than it would appear to be at first glance; two OCLC developers served informally as technical consultants for troubleshooting issues with XML, XSLT, CSS, HTML, and Javascript during the teaching of the course detailed in this chapter.

hand in shaping either the ideas or the language of any aspect of the legislation at any stage of the process? In simpler authorial contexts, too, texts and ideas proliferate in digital settings with remarkable speed. Jim Ridolfo coined the term "rhetorical velocity" to describe the "rapidity at which information is crafted, delivered, distributed, recomposed, redelivered, redistributed, etc., across physical and virtual networks and spaces." This phenomenon goes beyond mere virality and the inchoate hope (or trepidation) behind it, which proposes passivity where Ridolfo and DeVoss see agency in rhetors "composing for strategic recomposition" and actively grappling with potential reuse scenarios. These counterexamples suggest that the echoes between coding and writing point more to the differences between how they have traditionally been taught and how coders and writers compose in casual and professional settings. An intersectional approach to media-specific analysis helps to specify the nature of the media in question, for example, as code and corporate product, text and tweet.

Markup language presents a unique case study because, in practice, it also contains a form of writing. It is an encoded-text form in which code and writing are mutually supportive, neither disentangled from nor lesser than the other. N. Katherine Hayles describes this effect as "intermediation" via the work of Nicholas Gessler: "interactions between systems of representations, particularly language and code...between modes of representation, particularly analog and digital," and between human and computer agents, particularly at the point of the interface (*My Mother Was a Computer* 33). Importantly, markup language in the pedagogical milieu of the writing class presents a conspicuous testing ground for the central thesis of Vee's work: just as writing changed our understanding and practice of speech, coding changes our understanding and practice of writing. It also offers a concrete, discipline-specific response to the

question that drives this dissertation: what is the value of code for education, inquiry, and knowledge-making in the humanities?

As we count code among the available means of persuasion, decisions to use it or other modes such as writing are not so much the default as they are a deliberate choice informed by rhetorical and instructional needs. This is not without precedent in composition studies where the advent of widely accessible digital composing platforms has seen a concomitant culture shift towards design-stage inquiries regarding why one has chosen traditional writing assessments (or other genres, forms, and modes of composing) as the means by which students will work towards learning outcomes. For the first-year writing class I discuss in this chapter, I chose code not only for one or two assignments, but as the encompassing medium for all coursework; this choice was informed by a reflection on the goals of first-year writing in terms of both disciplinary thinking and the curricular ubiquity of the course in most students' programs across the disciplines. XML or extensible markup language emerged as the coding language best aligned with the spirit and aims of first-year writing, especially by foregrounding metacognitive and rhetorical thinking and scaffolding all writing activities in strikingly visible ways.

It's important for my case study, then, to consider XML and markup languages in both historically situated and technical terms. What is the context of their development and uses? What are their capabilities and limitations? How does XML work, and who are the stakeholders that develop standards for its use in scholarly contexts? As an instance of both text-based writing *and* code, XML lends unique insight into Vee's sense of grammatization vis-a-vis Bernard Stiegler: the process by which information is abstracted, discretized, reproduced, and scaled from speech, to writing, to code. Walter Ong, for example, argues that the historical invention of

writing "initiated what print and computers only continue, the reduction of dynamic sound to quiescent space, the separation of the word from the living present, where alone spoken words can exist" (81). Though Ong ultimately views writing with nuance and even praise for its effects on human consciousness, his ambivalence shows how grammatization often suggests alienation and loss. Even further, Vee explains, critics warn of rationalizing technologies that atomize and datatize Ong's 'living present' or what John Tinnell evokes as the "continuous flux" of experience (136).[83] In other words, critiques of writing technologies in this vein contend that the more the linguistic act can be reproduced, manipulated, transformed, taken apart and put back together, filtered and reduced, and so on, the less it 'means' in terms of authentic and fully contextualized communication. It is a critique not entirely dissimilar to Walter Benjamin's in "The Work of Art in the Age of Mechanical Reproduction," whose central thesis involves the "decay of the aura" of singular artworks (222).

Yet, as Vee takes from Heidegger, technologies also reveal new ways of thinking, and she favors a neutral sense of affordances and constraints over a postlapsarian narrative of decline (118-20).[84] Venturing further than Vee's middle-of-the-road correction to Ong's and others' suspicious view of writing technologies—in short, that technologies offer *different* scenarios

[83] Though he does not mention it by name, Steigler alludes to the Holocaust as the "shadow" of the Enlightenment and "rationalisation of the world" via Adorno and Horkheimer's writing circa 1944. Cynthia Haynes, too, finds the Holocaust lurking in the logic of datatization and casuistry vis a vis the Hollerith tabulation machine, which was used by the Nazis to maintain what was essentially metadata about prisoners at the concentration camps (229).

[84] It is unclear to what degree Vee considers code as able to *add* information and contextual specificity in the face of predominant theories of grammatization that propose "a hierarchy of abstraction." She seems to view the semiotic continuum of speech, writing, and code more as a zero-sum differential—or Stiegler's notion of the *pharmakon* with its duality of medicine and poison—by which each semiotic system gains some affordances but loses others, offers some solutions but also presents challenges. Thus, while code may be an abstraction of writing, in Vee's thinking, it does not represent any less valuable an opportunity to exercise literate and creative agency (122-23). Vee focuses on programming overall, of which markup is a relatively small slice of the whole. Markup and XML in particular likely represent more of an exception rather than a counternarrative to Vee's Heideggeran brand of grammatization.

with no net loss or gain in the capacity to convey information and enact forms of thinking—I contend that XML allows us to *add more* context and information than would be available in writing alone. J. D. Applen describes XML precisely in this way, as "not just data, but data with context," which presents technical communicators with the opportunity to exert agency as knowledge managers in their professional settings (307). Used strategically, this layer of metadata serves the first-year writing class well by integrating critical self-reflection throughout the writing process, a reflection often deferred to coda-like activities that rely on haphazard recollection and tend not to bear as much curricular urgency despite their well-documented importance for the learning process. Overall, the intervention of this case study suggests the value of code not only for writing instruction and humanities education but also as an amendment to how we theorize the development of writing technologies with respect to how they represent, reconstitute, and reveal information and ideas.

<Markup Languages and XML/>

Stiegler, the theorist of grammatization, extends the scope of Derridean grammatology from language to technologies and emphasizes processes of change over systemic hierarchies. In examples as seemingly unrelated as the shift from pictographs to phonetic letters, the standardization and scaling of industrialized production, and the cracking of the human genomic code, "life in the world—human becoming, historical change, social organization—[is] the evolution and play of *gramme*," the foundational unit of measurement for grammatological thinking. The gramme "can include[s] all manners of technical gestures that maintain their

iterability and citationality apart from an origin or any one particular context" which would otherwise evince an irreducible "material, sensory, or symbolic flux" (Tinnell, "Grammatization" 135-36, emphasis original). We see an analog to the gramme via Paul Caton's use of the token, the basic unit of programming languages, in his proposal for a "pure transcriptional markup." Following the thesis that "at any one level, a token instantiates a single type," Caton's approach evacuates the parsed character data (i.e., encoded text) normally contained within elements into the attribute value fields of the elements themselves. All basic tokens become empty elements that are decidedly more gramme-like in their granularity. In his example, the code snippet `<persName>Joe</persName>`, which is otherwise perfectly acceptable TEI markup, becomes:

```
<word designation="persName">
    <character type="j" form="majuscule"/>
    <character type="o"/>
    <character type="e"/>
</word>
```

The search for an unambiguous level of granularity at which tokens instantiate only a single type in transcriptional markup—in essence, the pixelation of textual data—recalls the discretization of the grammatization process as Steigler imagines it. John Tinnell's explanation of Stiegler's theory is of interest for its invocation of composition studies as a foil for a bigger-picture take on the use of technologies in pedagogical contexts. Stiegler's work on grammatization counters the popular notion that technologies augment human agency as "prosthetic extension[s] of innate human capabilities" ("Grammatization" 137). Compositionists erroneously assume that digital technologies augment the student writer, Tinnell explains, in what he somewhat pejoratively

calls "digital tools scholarship."[85] He urges the field to pivot from questions about "how 'digital tools' can be used by student writers" towards a high-level inquiry into "how writing has been, is, and will be transformed by the global proliferation of digital media" ("Post-Media" 123-24).

Tinnell invokes grammatization to remind us that when we consider how a writer might use a new tool or technology, we must also robustly theorize how the medium (re)constitutes both the writer and writing itself. True to Stigler's notion of writing technologies as *pharmakon*, a Janus term signifying poison and medicine, Tinnell warns of the "general proletarianization of the consumer's existence" that follows from a "hyperindustrial investment in digital machines": in other words, a grammatization of the user experience similar to the division of labor and the hierarchies of information and decision-making agency in industrial settings ("Grammatization" 137). As medicine, however, "a new gramme is what transforms the medium into an everyday social practice, a cherished mode of aesthetic experience, and an essential vehicle of deliberative democracy" (145). I approach the use of markup language in the first-year writing class with such a sensitivity; rather than make a fetish of novelty that assumes that the intellectual work and agency of writing remains essentially unchanged albeit enhanced, I consider how students engage in a different kind of thinking and composing as a result of the shift from traditional writing (e.g., typing in/on a document file) to the dual-coding milieu of an XML editor. Against the notion that grammatization inherently involves a kind of information loss via abstraction and discretization, I propose that XML allows student writers to further contextualize their thinking. The reconstitutions of XML transformations, too, lend a fuller insight of a student's oeuvre over

[85] Similarly, Cheryl Ball critiques the lack of theoretical rigor in writing and presentations that focus on what she describes as "Here's What I Did in my Classroom one-off scenarios" involving the use of digital tools.

the course of the academic term. Leveraging XML in this way amounts to an iterative, additive process rather than one in which each process replaces another. Students write and encode their writing, all the while considering the full range of perspectives that the transformations afford.[86]

Among the many definitions of the Ancient Greek *γράμμα* or *grámma*, the etymological basis of Derrida and Steigler's gramme, we might venture both *mark* and *writing*. Markup, of course, predates modern computers. It may be traced to "the proofreader's [handwritten] marks on a manuscript" as well as "typographic and design information" added to a manuscript prior to typesetting (Spring 111). Historical perspectives tend to emphasize the presentational aspect of markup as instructions for how a text ought to look once it is printed on the physical page or displayed on a screen. Michael Spring's examination of the origins of copymarks for digital publishing, for example, focuses on "how the printable information within the file should be displayed," while Allen Renear's later introduction to descriptive markup in the *Companion to Digital Humanities* maintains its origin in instructions for "how something is to look in print" (111). Philosophically, however, I draw from older traditions of marginalia and annotation that focus on the intellectual content of a text rather than its appearance, such as the ancient scholia or medieval glosses. In a modern sense, I would expand N. Katherine Hayles's analysis of hypertext to include XML for its capability to incorporate adjacent discourses and linked data. Composing in XML, then, is a process of self-commentary with respect to the rhetorical work of the writing class, from matters of structure and organization to persuasive strategies attuned to particular rhetorical situations. More advanced writing courses may have the capacity to place

[86] *Transformation* resonates for grammatology and grammatization as well as for the more technical process of transforming XML into presentation-ready markup language such as HTML. That process, detailed later in this chapter, is called extensible stylesheet language transformation (XSLT).

the presentation of encoded text within the purview of student deliberation but given the already-crowded nature of the first-year writing course, I directed students' effort solely towards descriptive as opposed to procedural markup. In other words, students did not engage in coding the transformations that rendered their XML corpus file into an HTML-based website.

Renear identifies the late 1960s as the time when descriptive markup began to emerge as a desirable standard for electronic text publishing and processing. Over time, Renear explains, the practical benefits of descriptive markup were met with a growing sense that it offered a theoretically robust "model of text" by the late 1980s. According to Renear,

> The model in question postulates that text consists of objects of a certain sort, structured in a certain way. The nature of the objects is best suggested by example and contrast. They are chapters, sections, paragraphs, titles, extracts, equations, examples, acts, scenes, stage directions, stanzas, (verse) lines, and so on. But they are *not* things like pages, columns, (typographical) lines, font shifts, vertical spacing, horizontal spacing, and so on. The objects indicated by descriptive markup have an intrinsic direct connection with the intellectual content of the text; they are the underlying "logical" objects, components that get their identity directly from their role in carrying out and organizing communicative intention. The structural arrangement of these "content objects" seems to be hierarchical — they nest in one another without overlap. Finally, they obviously also have a linear order as well: if a section *contains* three paragraphs, the first paragraph *precedes* the second, which in turn precedes the third. (emphasis original)

A hierarchical model for text has not gone without challenge over the years; standoff markup, for example, sidesteps hierarchy for embedded pointers to externally stored data in what is perhaps a

more pronounced form of grammatization. Nonetheless, Renear emphasizes the flexibility of the hierarchical model with respect to genre and purpose. Descriptive markup specifies a structure and syntax but not the content itself; certain tags will be more useful for a handwritten letter, for example, than for a playtext. Different projects, too, may be interested in the a text for more than one reason; a team might encode a manuscript of a poem, for instance, to track the allusions to earlier literary works while another team might attend to the composing process evident in the physical state of the text via erasures, emendations, interlinear additions, and different hands (i.e., writers). As an *extensible* markup language, XML furthermore allows for the definition of new tags, attributes, and values to fully account for the nature of the texts at hand as well as the project goals.

While the advantages of descriptive markup became clear, the need emerged for a way to define these languages for machine and human interoperability. Descriptive markup would not be very useful if it were siloed across idiosyncratic implementations incapable of 'talking' to other projects and stakeholders. Developers in the Graphics Communication Association and the American National Standards Institute sought to address this problem across two decades, and in 1986 the International Organization for Standardization consolidated this work in Standard 8879 establishing Standard General Markup Language (SGML).[87] As a metalanguage SGML focuses on the grammar, lexicon, and other conventions of the languages it defines: for example, the characters that indicate the presence of tags, the tags themselves, the structure of tags and other components, and so on. In this way SGML allows software to 'understand' markup languages that it would otherwise not be able to read. Yet, as Renear relates, the promise of SGML initially

[87] www.iso.org/standard/16387.html

was stymied with the rise of WYSIWYG (what you see is what you get) word processors and desktop publishers. (Stielger would perhaps see this as a proletarianization of computer-based composing.) With the advent of the world wide web and the hypertext markup language (HTML) on which it was built, however, markup languages finally drew widespread attention as well as scrutiny. The web's meteoric rise meant that early HTML proved chaotic, inconsistent, and unsophisticated; a more sustainable and extensible way of encoding data was needed.[88] XML was developed in the late 1990s by the World Wide Web Consortium to address these problems and streamline the SGML framework. Renear writes that the most consequential improvement on SGML was XML's requirement that markup be "well-formed," which includes criteria such as the strict adherence to ordered hierarchy (i.e., no overlapping tags), the inclusion of closing tags for all open elements (previous approaches to markup allowed for the haphazard inference of absent closing tags), and the consistent application of letter case and quotation marks. Thirty years after the first forays into descriptive markup language, XML presented a compelling realization of the possibilities of modeling texts in digital environments.

At a basic level, XML specifies a syntax and structure for encoded text, meaning that it requires a particular form and arrangement for the code but does not mandate the vocabulary that coders use to describe texts. Elements are the basic unit of XML; elements may have any number of attributes, each of which may be assigned one or more values. Below is an excerpt from an instructor-generated example assignment in my markup-based writing class, the primary source analysis:

[88] According to Renear, early HTML (a) lacked any document type definition (DTD) that would standardize its use, (b) "indiscriminately" mixed descriptive and procedural markup, (c) lacked stylesheet provisions, (d) was not validated by web browsers, and (e) possessed an "impoverished" set of elements incapable of addressing the needs of specialized markup and web publishing projects.

```
<seg type="thesis">
    <seg type="thesis_part" n="1">
        <title level="a">Summer Arm</title> therefore consists both compositionally and
        conceptually of three major parts: the mechanical, the human, and the natural, a triad
        that the ParkeHarrisons have made the focus of most, if not all of their work.
    </seg>
    <seg type="thesis_part" n="2">
        In addition to raising questions about the nature of each of these parts, <title
        level="a">Summer Arm</title> leads us to reconsider how much they feed into or push
        back against each other.
    </seg>
</seg>
```

The first major assignment of the semester, this paper asked students to analyze a primary source (an artifact of any medium) using techniques described in David Rosenwasser and Jill Stephen's *Writing Analytically* with the goal of supporting an overall claim about what the source does and how it 'works.' The excerpt above demonstrates how students were to tag their thesis statements; because a topic of study was how thesis statements can have multiple parts or steps, students also tagged each of these components within their larger theses. Figure 9 below shows the thesis tags in the nested container format that is a hallmark of XML's ordered hierarchy.

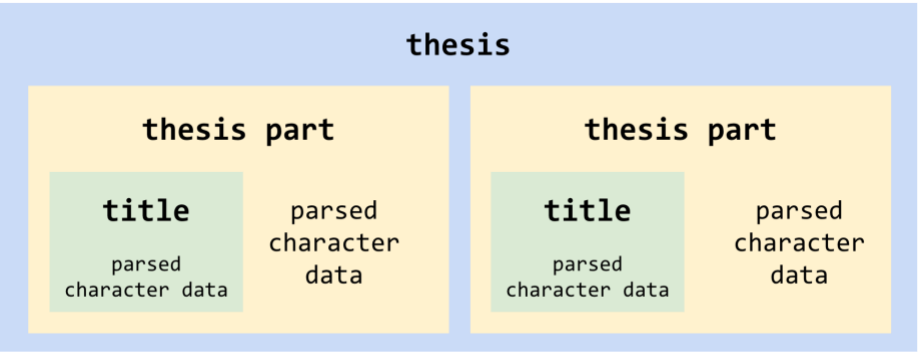


Figure 9: visual model of XML's nested hierarchies in the thesis excerpt of an example assignment

Using the generational language common to code, we see that the `<seg>` or segment element for the overall thesis (designated with the attribute `@type` and value "thesis") is a *parent* to additional segment elements designated with the same `@type` attribute but the value "thesis_part." The TEI

Guidelines define <seg> as an "arbitrary segment...of text below the 'chunk' level." Each child element in turn contains parsed character data—essentially normal text—along with the <title> element, which indicates the title of "any kind of work," according to the Guidelines. In this case the work is Robert and Shana ParkeHarrison's "Summer Arm," a visual piece that combines set design, photography, and digital manipulation.[89] Parsed character data is also included within this element (i.e., the text of the title itself), while the attribute @level, which the Guidelines define as the "bibliographic level," is given the value "a" for "an analytic item, such as an article, poem, or other work published as part of a larger item."[90] This example should illustrate how XML works for specific moments of text encoding that are particularly salient for composition and the goals of first-year writing. Leading students to a more nuanced understanding of the thesis articulation, for instance, is a perennial priority for writing instructors and core writing curricula.

The well-formedness of XML becomes a pedagogical affordance encouraging analytical and metacognitive thinking, which dovetails with the importance of well-formedness for a code text's readability, functionality, and interoperability. In their introduction to text encoding, Julia Flanders, Syd Bauman, and Sarah Connell confirm Renear's sense that well-formedness is the basis of XML's value proposition—it is "difficult to overstate [its] necessity," they write—while the worth of validity "depend[s] on circumstance" ("Text Encoding" 109). Validity is measured by the degree to which an XML file adheres to established vocabulary, syntax, and grammar as

[89] See www.parkeharrison.com/bodies-of-work/counterpoint-gray-dawn/284. "Summer Arm" is part of the artists' *Counterpoint Gray Dawn* series.

[90] Other values for @level are monographic ("m"), journal ("j"), series ("s"), and unpublished ("u"). The Guidelines indicate that if <title> is nested within the <analytic> element, no @level with a value of "a" is needed because its parent element already indicates this information. See tei-c.org/release/doc/tei-p5-doc/en/html/ref-title.html.

defined in a schema file or document type definition file. In the "Summer Arm" excerpt, for example, arbitrary spans of text are tagged with the <seg> element as opposed to, say, a or <section> element. The <seg> element contains other elements of the same type, but it does not contain a paragraph <p> element: something that it can be *contained by* but not itself contain. In this theory of the text a segment exists only below the paragraph level of organization. Above the paragraph level the or "text division" element is allowed because it points to "a [larger] subdivision of the front, body, or back of a text."[91] In the methodological framework of rhetoric and writing studies, as well as English studies broadly, the <seg> element's place and function in the TEI Guidelines' schema aligns more with close reading, while the element is poised to address a text's macrostructure, e.g., introduction, literature review, discussion, or, at a more fine-grained level, perhaps, individual 'points' or scaffolded subclaims within a larger argument. Put another way, we would not restrict such rhetorical and structural aspects of a text within the bounds of a single paragraph. Beyond technical and functional criteria, then, validity proposes a theory of the text and a way to operationalize the hermeneutical and rhetorical methodologies of the writing class.

<XSLT and the Text Encoding Initiative/>

Well-formed and validated XML can be manipulated and transformed for any number of purposes, by any number of people beyond the original authors, into other coding languages. In the case of my first-year writing class, the students worked on a single XML corpus file while I

[91] See tei-c.org/release/doc/tei-p5-doc/en/html/ref-div.html.

transformed that file into an HTML-based website for students to review in parallel throughout the semester. This process requires a second code file written in extensible stylesheet language (XSL) which is applied to the base XML file(s) to produce a third file of any type: XML, HTML, KML, JSON, etc. The full process of applying an XSL file to transform one file type into another is known as an extensible stylesheet language transformation (XSLT). To continue with the thesis example above, part of the HTML view I generated for the class involved extracting all students' thesis statements and displaying them on a single webpage to compare and contrast. One way of generating this transformation is the following:

```
<xsl:template match="/">
    <html>
        <body>
            <h2>Primary Source Analysis Thesis Statements</h2>
            <xsl:for-each select="TEI//div[@type='psa']">
                <p><strong><xsl:value-of select="head[@type='student_name']"/></strong></p>
                <p><xsl:for-each select="div/p/seg[@type='thesis']">
                    <p><xsl:apply-templates/></p>
                    <xsl:for-each select="seg[@type='thesis_part']">
                        <p><strong>Part <xsl:value-of select="@n"/></strong></p>
                        <p><xsl:apply-templates/></p>
                    </xsl:for-each>
                </xsl:for-each>
            </xsl:for-each>
        </body>
    </html>
</xsl:template>

<xsl:template match="title[@level='a']">
    <em><xsl:apply-templates/></em>
</xsl:template>
```

At first glance, XSL appears more impenetrable than XML because it is a procedural language as opposed to a descriptive one; not only does it tell a code editor what to do in order to generate a new file but it also must locate the data in an existing XML file. Flanders, Bauman, and Connell note that XSL " 'thinks' in XML" in the sense that it looks for relationships among hierarchical

nodes in order to transfer and transform the relevant data in the relevant nodes ("XSLT" 271). The excerpt above demonstrates three important actions in XSL: template matching/application, for-each loops, and value output. Below is the same excerpt with explanatory comments in green, using the conventional character notation for commentary in XML and XSL.[92]

```
<xsl:template match="/">
<!-- The single slash in the value field for the @match attribute refers to the "root" node of
the file, which contains everything. In TEI XML, the root element is <teiCorpus>. -->

    <html>   <!-- This tells the compiler that the following output will be in HTML. -->
        <body>
            <h2>Primary Source Analysis Thesis Statements</h2>
            <xsl:for-each select="TEI//div[@type='psa']">
            <!-- For-each loops cycle through matching elements: in this case, a text division
            <div> element with the attribute @type and value "psa" for the first assignment,
            the primary source analysis. Each student composed in their own <TEI> element,
            which is essentially a basic unit of organization for the TEI corpus file in XML.
            XSL navigates the XML document with the use of slashes in the "select" value. The
            double slashes here indicate that multiple levels or "generations" of nodes may be
            skipped, while a single slash strictly indicates a parent-child relationship. -->

                <p><strong><xsl:value-of select="head[@type='student_name']"/></strong></p>
                <!-- The value-of function selects particular data values. In this case, the
                value is output in an HTML-based paragraph in bolded text. From the @select
                attribute we see that the compiler looks within each primary source analysis
                <div> for a <head> element with the @type value of "student_name" in order to
                display the student's name above their thesis statement. -->

                <p><xsl:for-each select="div/p/seg[@type='thesis']">
                <!-- There are three nested for-each loops in this excerpt. The first loops
                through the base <TEI> elements to identify each instance of a primary source
                analysis for each student. The second loop, i.e., this loop, identifies each
                instance of the overall thesis within the individual primary source analysis. It
                is unlikely that students would tag a thesis more than once, but coding this as
                a for-each loop rather than a value-of output ensures that all instances of <seg
                type="thesis"> are output. -->

                    <p><xsl:apply-templates/></p>
                    <!-- The apply-templates command outputs any text or other content contained
                    in the selection at hand with specifications or modifications defined by any
                    template elements in the XSL file. For example, the template below includes
```

[92] XSL and HTML elements are distinguished with lighter and darker blue, respectively. XSL compilers distinguish elements by their namespace: the markup language to which the element belongs. For a succinct explanation that is friendly to the lay reader, see Flanders, Bauman, and Connell, "XSLT: Transforming Our XML Data," pp. 269-70.

```
                quotation marks around any analytic-level titles. If there are no templates,
                parsed character data will still be output for <xsl:apply-templates>. -->

                <xsl:for-each select="seg[@type='thesis_part']">
                <!-- This is the third for-each loop, operating within the overall thesis
                segment element. It cycles through each instance of a <seg> element with the
                @type value of "thesis_part" in order to output the thesis broken into what
                the writer has tagged as its component parts. -->

                    <p><strong>Part <xsl:value-of select="@n"/></strong></p>
                    <!-- Outputting value of the @n or number attribute allows for ordinal
                    listing of the thesis parts. Without the @n attribute, a more complex
                    XSL code sequence would be needed to define and output variables that
                    iterate with a for-each loop's cycles. In this case it is more expedient
                    to define the n-value for each thesis-part segment in the XML file. -->

                    <p><xsl:apply-templates/></p>
                </xsl:for-each>
            </xsl:for-each>
        </xsl:for-each>
        <!-- Here we see the nested for-each loops closing in a cascade. -->

    </body>
  </html>
</xsl:template>

<xsl:template match="title[@level='a']">
<!-- Templates apply to the entire XSL transformation regardless of their location; even
though this template is not part of the root template above, it still will apply to the HTML
output whenever the apply-templates command is given. This template applies double quote marks
to <title> elements at the analytic level, e.g., an individual work within a body of work. -->

  "<xsl:apply-templates/>"
</xsl:template>
<!-- To save horizontal space here, I have not included <xsl:stylesheet>, the root element of
an XSL file. All the code in this excerpt would be contained by this element. -->
```

The comment lines above explain the step-by-step of the XSLT at a granular level. An overall narrative of the transformation (or pseudo-code) might say: the XSL looks for each instance of the primary source analysis in the corpus file, and for each of those assignments, it prints the student's name and cycles through the text segments to output the thesis in its entirety and in numbered parts. The process by which the XSL navigates the XML nodes is called XPath, which

allows a coder to express instructions in machine-readable commands as shown in the 'pseudo code' (i.e., the narrated actions of the code) and actual code below.

> *At any point in each `<TEI>` element look for a `<div>` element that has an `@type` value of "psa." Once there, go one level down and output the value of the `<head>` element with `@type` value of "student_name." Do this for all `<TEI>` elements and output the names in a vertical list.*

```
<xsl:for-each select="TEI//div[@type='psa']/head[@type='student_name']">
    <xsl:apply-templates/><br/>
<xsl:for-each/>
```

The HTML-based output of all thesis statements provides a striking and generative prompt for comparative discussions about thesis construction and the overall goals of the analytical essay.[93] Broadly, this example adumbrates the degree to which XSLT allows for ways of imagining and reflecting on XML-based student writing as it is filtered, (re)arranged, and manipulated—a meaning-added rather than meaning-detracted exercise contra the claims of grammatization. It allows students to consider their work at all scales: a single assignment by one student, all writing from a particular student, the collective work on an individual assignment, and the collective work on all assignments throughout the semester. For my first-year writing course students did not code the XSL transformations nor did they delve into the mechanics thereof beyond a cursory sense of how I was generating the HTML-based website. As an instructional design decision I was mindful of matching complexity with course level and avoiding undue

[93] The XSL editor allows for more languages than HTML. For example, files may also include cascading stylesheet (CSS) language and JavaScript, both of which are critical to a webpage's presentation and compatibility across software and devices. For an extensive guide, see *XSL Transformations (XSLT) Version 3.0*, edited by Michael Kay, 8 June 2017, W3C, www.w3.org/TR/xslt-30/.

burden on cognitive load amid an already busy curriculum. A different course, or perhaps even the same course scaffolded differently, may very well benefit from students taking on decisions about the transformation process, especially in the spirit of open pedagogy and participatory design. Full technical knowledge would not be needed for high-level discussions about the possibilities enabled by the malleability and manipulability of XML—i.e., grammatization—as well as the potential uses of their work within or beyond the context of the class, and the ways in which the transformation process informs or alters their sense of the practices and purposes of writing.

To transform XML successfully, one must be familiar with the conventions that the XML follows, i.e., its validity. While validity may be relatively less important than well-formedness overall, it becomes much more significant for projects that seek to model texts and writing with rigor. If we take seriously the challenges of sustainability, usability, and transparency for digital projects, validity invites an ongoing consideration of the balance between standards for markup's lexicon and grammar with flexibility that follows from the spirit of extensibility, i.e., the ability to alter and add to standards in a project's best interest. It would be prohibitively time consuming to code transformations if every project took an idiosyncratic approach to validity. In fact, validity works similarly to genre in the sense that it relies on document type definitions and schema that specify form(s) and content. Because of markup's traditional use in English studies for the encoding of literary and historical texts, scholars working in the wake of markup standards development in the 1980s faced the challenge of creating data that retained value beyond a project's immediate output such as a file or other digital object. Would researchers be able to understand, use, and adapt the encoded text and metadata? How could the encoding

methodology reflect the same intentionality and ethos as the curatorial, editorial, analytical, and hermeneutical work of scholarly editing and publishing?[94]

A year after the ISO released standards for SGML, a meeting of the Association for Computing in the Humanities addressed the dearth of interoperable standards in "the encoding and exchange of literary and linguistic data," culminating in the adoption of what are now called the Poughkeepsie Principles for their convening location at Vassar College (Ide and Sperberg-McQueen 5).[95] These principles laid the groundwork for the organizational development of the Text Encoding Initiative—a collaboration between the ACH, the Association for Literary and Linguistic Computing, and the Association for Computational Linguistics—as well as for the development of the TEI Guidelines themselves. Despite the spirited work of its first decade, by the late 1990s the TEI had become an unfunded effort while many of its original developers had shifted their attention to other work. On the precipice of obsolescence, the TEI was reinvigorated with what founding editor Lou Burnard describes as a "management buyout" that resulted in the creation of the TEI Consortium: "a democratically constituted, academically and economically independent, self-sustaining, nonprofit organization." One of the first acts of the Consortium was to release the fourth iteration of the Guidelines, this time in XML rather than SGML. Burnard relates that by the 2007 release of the P5 Guidelines the TEI had "consciously transformed itself into a classic open source project" that had become "reliant on community input both to define

[94] In their historical survey of the TEI, Nancy M. Ide and C.M. Sperberg-McQueen write that "[b]ecause of the lack of a unified, standard format, scores of such encoding schemes were developed from scratch or adapted from existing schemes in the 1960s, '70s, and '80s. These schemes typically reflected the specialized interests of their originators and were, by and large, incompatible; the end result was that a text encoded for one purpose or piece of software often required substantial editing to be used for another purpose or with other software, if it was reusable at all" (5).

[95] The nine Principles are included in full in TEI document EDP1 at tei-c.org/Vault/ED/edp01.htm.

and to execute all of its future development." For its longevity and ongoing relevance, Burnard writes, it may be less accurate to describe the TEI as an academic or community research project than to describe it as an open-ended activity" or even "a research infrastructure" for projects and activities. Indeed, the Guidelines have become a lingua franca among literary, textual, linguistic, and historical scholars for the sustainable development of datasets and research production in the digital humanities.

The TEI Guidelines are substantial; released on 25 February 2021, version 4.2.0 of the P5 Guidelines (revision 736c0acf0) spans 2000 pages in PDF form.[96] Readers may also browse the website version of the Guidelines with appendices for the 589 elements from `<ab>` to `<zone>` and 268 attributes from `@absolute` to `@xml:space`.[97] Revisions to the Guidelines, which occur twice a year, may involve the addition of new elements, attributes, and classes (groupings of elements or attributes that address a common task or issue), revision and restructuring of existing technical content in the Guidelines, improvements in the written explanations and examples, housekeeping changes, and updates in accompanying files such as stylesheets for processing and conversions.[98] This is an impressive scholarly endeavor in its own right, and as James Cummings (TEI Board of Directors 2020-23) reminds us, the community's stewardship of the Guidelines also provides (1) a rich historical record of how scholars have modeled textual phenomena with/in code, (2) an evolving "consensus-based" methodology for operationalizing those models, (3) more paths for

[96] The latest version of the Guidelines are available at tei-c.org/release/doc/tei-p5-doc/en/Guidelines.pdf.
[97] As of revision 736c0acf0, 25 Feb. 2021.
[98] Aside from the more technical or minor updates of version 4.2.0, the TEI introduced the `<persPronouns>` element for "the personal pronouns used, or assumed to be used, by the individual being described," as well as a nested trio of elements—`<ruby>`, `<rb>`, and `<rt>`—for ruby annotations, "a particular method of glossing runs of text which is common in East Asian scripts." Early reactions to these additions praised their multilingual, inclusive, and queer resonance, demonstrating the larger implications of the TEI's technical infrastructure for modeling texts and other humanistic data (@amclark42; @quinnanya).

custom schemas, (4) a mechanism for optimizing the interoperability of research data, and (5) a documented format that supports the goals of archival preservation ("World of Difference" i58). The TEI and the Guidelines are not without their enduring myths and misconceptions, however, emerging in part from the tensions between comprehensive and limited coverage (e.g., there are either too many or too few elements), flexibility and precision, specialized and casual expertise, fixity and emendation of standards, and novel developments and aging infrastructures for the encoding and management of textual data (i58-i59). For the purposes of this chapter, the most salient misconception that Cummings names is that the TEI is useful only for the production of digital scholarly editions. While he focuses on alternative output such as "supplementary files, indices, databases, interactive visualizations, glossaries…bibliographies, working papers, meeting minutes, and slides for lectures, as well as other teaching materials" along with "[large] text-bases for querying and analysis" rather than publication-oriented data, we can also imagine an expansion of the TEI's purview from texts originally composed in some other medium to texts composed deliberately in TEI XML not only for its interoperability and transformation but also for the kind of thinking that it requires as a framework for composition (i74). Cummings gestures to the TEI's pedagogical potential beyond the short-term, intensive workshops that tend to focus on compliance-based outcomes. "As part of that teaching, it is sometimes easier to focus on the facts as instantiated by rules," he writes, "rather than explaining the underlying concepts of why something is being encoded in this manner. Ideally TEI pedagogy would have the leisure to explain both the precepts (the 'how') and the underlying concepts (the 'why')" (i75-i76). So too with writing, and hence the alignment between TEI XML and the most challenging but valuable goal of the first-year writing class: infusing metacognition throughout the writing

process. With the full semester to compose and encode, students have the curricular space to dwell in the why.

<Metadata and Metacognition/>

Metadata—data about data—usually is operationalized for bibliographical purposes in a humanities context. The most rudimentary appearance of metadata in the first-year writing class is the works cited list at the end of an essay with authors, titles, publisher information, dates, and other data. As a more technical example, every TEI XML file begins with a `<teiHeader>` element that contains a rich nested structure of descendent elements clarifying metadata for the encoded text. The TEI Guidelines describe the header as having five conceptual parts: a file description, an encoding description, a text profile, a file revision history, and data from non-TEI schemes if appropriate. These areas include information about the editorial and technical teams; the sources from which the encoded text was drawn (i.e., from a bibliographical perspective); "classificatory and contextual information about the text" including entries for all named entities and references in the encoded text(s); philosophical and technical approaches to the application of TEI markup including "whether (or how) the text was normalized during transcription, how the encoder[s] resolved ambiguities in the source, what levels of encoding or analysis were applied, and similar matters"; and a descriptive list of changes (attributed to individuals) at a level of granularity that is appropriate and feasible for the project.

Both the works cited list and the `<teiHeader>` cleave to the standard understanding of metadata as "systems to name, classify, and manage data" for purposes ranging from cataloging

to market research (Acker 322). Traditional understandings of metadata tend to place it before, after, or entirely separately from the object(s) that it describes. Yet, descriptive markup in the body of encoded text also offers metadata, whether *structural* such as division or paragraph, *rhetorical* such as passages that accomplish a particular persuasive task, *stylistic* such as emphasis via italics or underlining, *compositional* or *transcriptional* such as deletions and emendations in a draft or manuscript, and *contextual* such as editorial annotations and references to named entities. This metadata adds a metacognitive layer to the act of composing as a student encodes a narrative of what they're doing with/in their text. The thesis markup for "Summer Arm," for example, demonstrates these dual registers, which we might loosely compare with intradiegetic and extradiegetic narrative, if we translate the `<seg>` tags for `@type` "`thesis`" and "`thesis_part`" as a kind of declaration: "here I am articulating my thesis statement, and it is made up of two parts that build on each other like so."[99] If the tags had included `@function` attributes the metadata would also indicate what each thesis part addresses through a taxonomy of values such as descriptive and interpretive thesis types (e.g., "this portion of my thesis offers an authoritative description of the image's formal structure, and this portion offers an interpretation based on that structural definition"). With metadata that focuses on what texts *do* in addition to what they *are*, students engage in a writing process inflected with a greater degree of metacognition, which I define informally as an awareness of what they're doing, a reflection on why they're doing it, and an effort to connect those thoughts to course concepts and methods.

[99] Naomi Silver's description of metacognition further suggests the structural analogy to diegesis: "the moment of standing above or apart from oneself, so to speak, in order to turn one's attention back upon one's own mental work" (1).

Metacognition has long been a central concern of the literature on teaching and learning. It manifests across disciplines and levels of instruction as the ongoing practice of understanding how one is engaging in coursework, how well that engagement is working, and how the granular tasks and insights connect with the overall learning goals and intellectual project of a course and discipline. Surveying research since the 1970s, Naomi Silver finds that cognitive and educational psychologists focus on metacognition per se while scholars in other disciplines and experiential fields often invoke a looser framework of reflective practice (2). Though echoes and intimations of metacognition appear in the work of educational and psychological theorists from William James to Jean Piaget and Lev Vygotsky, John H. Flavell explicitly developed the concept across several publications in the later 1970s, providing the "bases for all of the major metacognitive schemas in circulation" (9). Flavell ultimately lands on a four-part framework for metacognition: knowledge, experiences, goals/tasks, and actions/strategies (906). The use of descriptive markup language in the first-year writing class, for instance, lies within Flavell's notion of experiences that attempt to induce "careful, highly conscious thinking" that can result in the refinement of the goals for the task at hand (i.e., the paper or project) and the writer's "metacognitive knowledge base," i.e., their familiarity with disciplinary methods and ways of thinking (908). In other words metacognitive experiences bolster both the short-term and long-term goals of first-year writing by leading the writer to consistently monitor and revise their approach to a particular assignment while also building on their understanding of writing frameworks, conventions, and strategies in and across the disciplines.

Silver indicates that transfer (the application of concepts, skills, and methods beyond the original context in which they were learned) presents "one of the important open questions in the

research" on metacognition. Researchers differ on whether and to what degree the type and scale of cognitive tasks influence the transfer of metacognitive practices across domains of knowledge. Importantly for this chapter, Silver emphasizes that precisely because of this ongoing inquiry we cannot understate the value of instructional guidance for students' ability to integrate experiences into their knowledge base and transfer it into new situations (10-11). Moreover, the cultivation of metacognitive practices is linked with engagement and self-regulation: two major predictors of student success across the disciplines (3-4). Research on self-regulated learning practices attends to students' planning, motivation, habituation, monitoring, and self-evaluation. Silver considers metacognition to be a type of self-regulation, while others consider it in parallel with motivation as the major reagents for the development of self-regulation mindsets and practices (Zimmerman and Moylan 299). Overall, the discourse on transfer acknowledges the importance of meta-goals beyond the acquisition of a particular set of content knowledge, procedures, and applications that add up to something of a provincial mindset when it comes to imagining what students will need from and do with their learning experiences in the long term. Aside from the idealistic critique of a vocational or career-training paradigm in higher education, research suggests that few students end up working in fields that are "directly related" to their college major (Abel and Dietz 8). All of this points to the significance of transfer and metacognition for long-term student success: the ability to apply reflective and analytical methods while adapting domain-specific knowledge to adjacent or even far-flung areas of practice.

First-year writing and other introductory or lower-level writing courses are uniquely implicated at the intersection of transfer, metacognition, and self-regulation given that many of these courses are designed as—or casually take on the role of—so-called 'service courses' whose

student body is composed mainly of non-majors in English or writing departments. The question often looms large for students and instructors regarding how the intellectual habits that these courses cultivate intersect with a wide range of majors, disciplines, and professions. The Council of Writing Program Administrators "Outcomes Statement for First-Year Composition" (v3.0) declares that "[a]s students move beyond first-year composition, their writing abilities do not merely improve. Rather, their abilities will diversify along disciplinary, professional, and civic lines as these writers move into new settings where expected outcomes expand, multiply, and diverge." The CWPA importantly leads us to consider students' intellectual and professional journeys as dynamic rather than static as they pertain to writing practices and perspectives; a history major may end up working as a product design engineer, for instance, where the writing practice is unique to the stakes, purposes, and audiences of written documentation in the field (Beaufort 133). The growth of technical and professional writing as a curricular and scholarly focus, for instance, attests to the disciplinary and institutional recognition that writing-based skills and careers require this kind of epistemic and rhetorical adaptation. Jody Shipka warns against "the myth that writing is a generalizable skill that, once successfully acquired, will serve students equally well irrespective of what they are attempting to accomplish." We should instead understand writing in terms of "flexibility, adaptation, variation, and metacommunicative awareness" (83). This approach implicates disciplinary differences as well as the multimodal turn in composition studies; Shipka urges against merely "equating multimodality with [producing] digitally based or screen-mediated texts" in favor of a pedagogy focused on reflective decision-making across modality, genre, platform, purpose, and format (84-85). A student's ability to meet these protean circumstances attests to their preparation and practice in

the way of metacognition: writing with writing in mind. Metacognitive strategies, too, "are not generic across subjects, and attempts to teach them as generic can lead to failure to transfer" (National Research Council 19). Scholarship on transfer in/from the writing class affirms the significance of a reflective practice and the mental habit of "*think*[ing] like writers" as students engage in "big-picture" comparisons with other writing situations (e.g., in other disciplines) as well as a general theory of writing that anticipates practice beyond classroom circumstances (Yancey, Robertson, and Taczak 4, emphasis original).

The use of descriptive markup infuses metacognition throughout the composing process. Conventional planning and reflective activities may isolate metacognition at the initial and final poles of an assignment sequence, while draft revision and peer critique often remain unrealized in their potential to leverage insights gained from sustained metacognition. Instruments such as Anne Frances Wysocki and Dennis A. Lynch's design plan framework, for example, encourage writerly metacognition but leave room for integration beyond invention and planning phases into drafting and revision.[100] Exploring metacognitive strategies for writing instruction, E. Ashley Hall, Jane Danielewicsz, and Jennifer Ware implemented Wysocki and Lynch's design plan "in a sustained, iterative manner throughout the composing process" in their first-year writing courses. Evoking the intersection of metacognition and self-regulation, they argue that a "systematic and iterative" approach to reflective and reflexive writing leads students to "monitor, regulate, and *improve* their composing choices" (149-50, emphasis original). They define metacognition for writing studies as "thinking about the decisions, choices, and use of intellectual strategies that

[100] See Anne Frances Wysocki and Dennis A. Lynch, *Compose, Design, Advocate: A Rhetoric for Multimodal Communication* (3rd edition), Pearson, 2018. A design plan template asks writers to consider purpose, audience, context, genre, media and materials, and arrangement.

happen before, during, and after writing." Of the four steps that make up this recursive process, descriptive markup provides a way of engaging students directly in the first two—"awareness of one's own thinking" and "articulation of the thinking process"—while also leading students to "reflection" and "questioning and challenging previous choices" (149). Beyond markup itself in my first-year writing class, the shared corpus file in the code editor and website transformations sought to cultivate reflection and self-questioning with latitudinal and longitudinal perspectives whereby students observed the development of each other's work as well as how their own had come together throughout the term. The encoding process and digital platforms, then, provided a four-pronged approach to metacognition through (1) the tagging of particular writerly moves and rhetorical aspects as part of the composing process, (2) perspective on how a student's thinking and work had developed throughout the class, (3) a sense of how peers likewise had developed their ideas and writing throughout the semester, and, as a result, (4) the application of insights gained from those observations and comparisons. Pedagogically speaking, the innovation in my experimental course design in large part concerns the scaling of metacognition from attention to decision-making during the act of writing to summative retrospectives, as well as the expansion of the metacognitive gaze from a writer's own work to that of their peers and even the collective corpus and intellectual journey of the class as a whole.

While the research on metacognition in writing studies is well developed, the categorical frameworks and instructional interventions tend to address student work at a scale of the overall project or even course. Gwen Gorzelsky, et. al., for example, draw from the grounded taxonomy of Brianna M. Scott and Matthew G. Levy to propose a vocabulary for metacognition attuned to

the unique needs of writing instruction.[101] The refined taxonomy uses the two categories common to the literature on metacognition—knowledge and regulation—to address a writer's knowledge of themselves (particularly *as* a writer), the nature of the task at hand, and the "approaches one might effectively use to complete a project." For a writer's regulation, Gorzelsky, et al. include the planning process, the monitoring of "one's cognition and efforts toward a project," the active choices and decision-making as a result of self-monitoring, and the assessment of "the quality of a completed project." To Scott and Levy's dyadic framework Gorzelsky, et al. add the synthesis category of "constructive metacognition," a transfer-oriented outcome of writing instruction that, by focusing students on the elements of metacognition, fosters "[r]eflection across writing tasks and contexts" and the use of "writing and rhetorical concepts to explain choices and evaluations and to construct a writerly identity" (226). However, while reflective thinking may play a central role in metacognitive theories and taxonomies, students' abilities to engage effectively in writing of this sort (and instructors' abilities to respond meaningfully) vary widely. Thomas Trimble and Adrienne Jankens applied Gorzelsky, et al.'s taxonomy to study students' final reflective writing with the goal of better understanding the rhetorical moves that students make and improving the quality and take-aways of the summative-reflective writing process. They suggest that reflective writing may itself better model metacognition by first inviting students to compose generative drafts that draw primarily from their experiences and emotions, followed by revisions that draw from Gorzelsky, et al.'s framework. This "[p]urposeful construction" of reflective writing leads instructors to understand

[101] See Brianna M. Scott and Matthew G. Levy, "Metacognition: Examining the Components of a Fuzzy Concept," *Educational Research*, vol. 2, no. 2, 2013, pp. 120-131.

student metacognition with more clarity while students "develop the constructive metacognition that will build their writing knowledge for future contexts" (450-51). It is worth noting that descriptive markup would be particularly suited to Trimble and Jankens's exercise by assigning tags—essentially, metadata *about* metacognitive moves—at the segment level of student writing, e.g., `<seg type="monitoring">`.

Summative-reflective writing assignments present students with a complex genre and a challenging task as they struggle to integrate piecemeal reactions into a systematic understanding of their intellectual journey throughout the course. The scale of analysis in these essays looks to the course or scaffolded project as distinguished from the granular, "in-process" moments of metacognition that themselves present the opportunity to scaffold up to a comprehensive inquiry (Trimble and Jankens 441). The problem can resemble the paradox of Achilles and the tortoise; considered as a distinct genre, the reflective essay occurs before or after but not concurrent with the writing itself. A markup-oriented approach to composing adds a layer of metacognition and reflection to the writing itself as students become better readers of their own work in-process. Moreover, they become attuned to writing as a dynamic process and rhetorically active medium; the (en)coding asks them to think about their rhetorical and compositional choices, as well as writing as an ongoing process and series of actions beyond the static content of what it is saying, i.e., writing-as-verb as opposed to writing-as-noun (Shipka 13). Crystal VanKooten has theorized this uniquely in-process approach to metacognition as "meta-awareness about composition," which she defines "as an ability to move consistently between enacting compositional choices and articulating how and why those choices are or might be effective or ineffective within a rhetorical context." From observing students during class meetings and follow-up interviews,

VanKooten proposes a four-part framework for understanding meta-awareness to better identify it in research and better cultivate it among students in composition-based learning environments. Across the taxonomy—process, techniques, rhetoric, and intercomparativity—the effect is "to move knowledge from the practical to the discursive, [to move] between actions and articulations more often or more consistently." In other words, meta-awareness is not just about knowing what one is doing; rather, it is about how one evinces that knowledge in discourse at strategic times during the composing process. For my XML-based first-year writing class, knowledge moved from the practical and, even further, the instinctual, into the explicit discourse of markup. Before detailing that approach further, however, it is important that I provide a detailed background for the curriculum onto which this pedagogical experiment was grafted. While some aspects of the first-year writing course are unique to the particular institution and time, it also is representative of the goals and format of writing courses at many U.S. institutions for its scaffolded assignment sequence, its focus on research skills and digital and information literacies, and its integration of multimodal communication and analytical heuristics.

<First-Year Writing at Ohio State Circa 2013/>

English 1110.01, the standard section of first-year writing at Ohio State, aimed for several goals serving a diverse range of students in the space of a single semester. The course focused on analytical and research-based writing as well as public and multimodal communication skills. The 'spine' of the curriculum, so to speak, was an analytical research project concerning a media artifact of the student's own choosing in consultation with the

instructor. The artifact could take any media and modal form: text, image, audio, video, as well as any combination thereof.[102] The research project consisted of five written assignments that progressed students through an inquiry regarding their chosen artifact. The first assignment, the primary source analysis, asked students to apply heuristic methods drawn mainly from the course textbook, David Rossenwasser and Jill Stephen's *Writing Analytically* 6th edition, in a "focused analysis" that sought "(1) to develop analytical claims about primary evidence and (2) to develop research questions that will yield compelling insights about primary materials in future writing."[103] This brief essay, in other words, engaged students in a close reading of their primary source that identified how it constructed and conveyed meaning with an eye toward larger questions that merited further research. (Appendix A(II) includes an example of a primary source analysis in XML.) Following an introductory unit on research methods and literacies, the second assignment asked students to put together a short annotated bibliography of sources that spoke to the questions they'd articulated in the primary source analysis. Returning to that essay, the third assignment asked students to integrate evidence and insights from at least one of the sources in their annotated bibliography. This revised version of the primary source analysis, called the secondary source integration, introduced students to analytical practices such as citing research,

[102] Instructors of English 1110.01 were encouraged to work with students to maintain a sense of scope for the chosen artifacts, i.e., that they would be self-contained enough to provide workable parameters for relatively brief analyses (the final paper, for example, required only 1750-2250 words). Textual artifacts tended to be of journalistic length, e.g., an opinion-editorial. Video artifacts tended to cover only a few minutes, e.g., an isolated scene from a film or a television show, a movie trailer, a music video. Audio artifacts likewise stuck to brief formats, e.g., a recorded song, a live performance. Students drew images from a range of contexts, e.g., an album cover, a playbill. Physical and virtual environments also provided options for artifacts, e.g., a building on campus, an app's user interface. Of the 24 students enrolled in the course, 12 (50%) chose video artifacts, 5 (20.83%) chose textual artifacts, 3 (12.5%) chose audio artifacts, 2 (8.33%) chose visual artifacts, and 2 (8.33%) chose environments as artifacts.

[103] Quotes here are taken from the syllabus as it was adapted from the first-year writing program's template in 2013. The language likely reflects a mix of that developed by program administrators led by Edgar Singleton along with my own minor edits to align the syllabus with my chosen theme, approach, and instructional tone/voice.

situating their writing among a larger discourse, and responding to others' findings and arguments with precision and nuance. The fourth and fifth assignments—a rough draft and a final draft—asked students to continue integrating sources and to make a fully substantiated claim about their artifact in a lengthier essay called the analytical research paper. In-class peer reviews and one-on-one conferences with the instructor supported students' drafting and revision at this stage of the project.

First-year writing and composition courses often organize student inquiry loosely around a theme chosen by the instructor of record based on their interests and expertise as well as their sense of larger exigencies and resonance with the cultural and political zeitgeist. English 1110.01 engaged in this practice at the time, and for my section I chose *codes* as a theme for its relevance to students' use of TEI-compliant XML as a compositional medium. Included in the syllabus was a statement explaining the capaciousness and flexibility of the course theme:

> Approximately twenty years after the globalization of the Internet in the early 1990s, we now take codes for granted in our daily lives. Seemingly fulfilling the prophecy of *The Matrix*'s endless stream of green characters, we continue to mediate our interactions with the world around us through technology driven by digital codes. Even human beings have been "decoded" as a genome, and in theoretical physics unified field theory and the theory of everything seek to reveal the code by which the universe operates.
>
> In this section of first-year writing, students' projects will explore the nature of codes, broadly considered. Beyond science and programming, codes are everywhere. As a student at Ohio State, you are held to a code of student conduct. Depending on where you are, be it a study hall, coffee shop, bookstore, bus, or bar, there are unspoken social

> codes regarding what is and isn't acceptable behavior. We turn to many authorities for the codes that govern us—law, literature, tradition, history, belief, credentials, expertise, etc. Language itself is a code of sounds and symbols, and we use different codes when talking to friends via text message than we do when talking to professional colleagues at an official function. While some codes are all-encompassing by design, others actively exclude through deliberate illegibility, whether they're the military codes protecting national security or the passwords protecting a child's clubhouse. Regardless of the situation, any code makes an argument about who we are, what we should be, and what we should be doing; we therefore should attend to it with the focused analysis that first-year writing seeks to foster. (Conatser, Syllabus for First-Year Writing)

Providing concrete paths of analysis rather than closing them off (i.e., rejecting lines of inquiry because they did not fit), the thematic approach assisted with one of the larger, often unspoken or unwritten goals of first-year writing: leading students to identify and engage with the implicit stakes of texts and discourse, the 'big ideas' that often lie behind that vexed and vexing response written on so many drafts: 'so what?'

In addition to the five-assignment sequence, which focused largely on writing skills and rhetorical analysis, students practiced public and multimodal communication through a parallel sequence of activities that led to a class symposium of oral-visual presentations using a modified PechaKucha format of exactly 15 digital slides with exactly 20 seconds of narration per slide.[104] In anticipation of the presentation, students completed five brainstorming and reflection activities

[104] Similar to TED Talk and Ignite presentations, PechaKucha promotes brief, compelling presentations that spur conversation, provoke thought, and combine entertainment with education. For more, see www.pechakucha.com.

with a sixth following the symposium. For the presentation, students could narrate their research process for the written assignment sequence, focusing on the development of their ideas about the primary source. Or, they could address the primary source itself, explaining a significant part of its meaning and implications for the course theme. Additionally, students could focus on some of the larger implications, stakes, or ideas to which their primary source gestures; for instance, a student who has chosen, say, an ad for the Make-a-Wish Foundation may deliver a presentation that meditates on the rhetoric, affect, and ethics of representations of youth and child mortality as they intersect with the spectacle of extravagant wish fulfillment. Student-audience members had four rotating (graded) roles during the presentations: (1) taking notes on a shared Google Doc on both the presentation itself as well as the audience questions and discussion; (2) leading the Q+A after each presentation with open-ended questions and observations; (3) writing a brief response to the presenter before the end of the class meeting; and (4) composing a longer response letter that reflected on the presentation and discussion holistically. These texts were composed either directly in the class corpus file or later transferred there among the presenter's section to preserve a full record of the symposium presentation as a public communication event.

Other in-class activities for first-year writing trend towards the productive, yet ephemeral moments that ask students to reflect on ideas and experiences, share insights, explore questions, participate in thought experiments, and make connections. For English 1110.01, daily activities were left to the instructor's discretion; of these, eight were composed in the class corpus file and took advantage of the affordances of XML, e.g., a revision activity that used `<choice>` elements to record all versions of modified passages in an example student essay. In addition to providing detailed curricular context for the learning environment in which students used markup language,

I have also suggested a limitation for the proof of concept in the complexity and intricacy of the curriculum prior to the additional, unfamiliar layers of XML, the TEI Guidelines, the <oXygen/> code editor, the corpus file, and the subversion process. Already adapting to the novel learning environment of the postsecondary writing class, already expending a good deal of energy to keep up with a busy sequence of assignments with unfamiliar names, terms, and expectations (as they also kept up with three or four other courses), students did not have an ideal amount of cognitive bandwidth to devote to the code-specific aims of the course. More than anything else, the choice of course and curriculum to implement the proof of concept was guided by what was available to the instructor at the time as a pre-candidacy graduate student in the department (whereas a more senior position may have had access and ability to negotiate a special topics course).[105] Ideally, the curriculum of an XML-based writing course would be developed hand-in-hand with the technical process and components. Nevertheless, the students benefited from the metacognitive focus of the markup as well as the radical transparency of composing all assignments on a shared space. Regarding the latter point, one of the major shifts for students beyond the markup per se was composing in the <oXygen/> code editor as opposed to a word processor—typically Microsoft Word but sometimes a platform such as Google Docs or even Apple Pages—with significant implications for workflow and their sense of the writing process.

<From Word Processing to Code Editing/>

[105] The fourth chapter details such an approach to designing and teaching a course, which I was able to arrange for as an academic staff member at a different institution. While this course was not necessarily a *writing* course per se, it nevertheless incorporated my own reflections, my students' feedback, and scholarly feedback on the writing class detailed in this chapter.

In educational contexts the word processing program has become so naturalized in the writing and assessment process it is difficult to disentangle as one possible platform among many options for composing.[106] The rise of learning management systems (LMSs) such as Blackboard, Canvas, and Moodle further entrenches our notion of the word processing file (docx) as the *telos* and main deliverable of writing-based assessment as it is downloaded, marked by the instructor, and re-uploaded for the student to review or graded directly in the online interface as the LMS likewise has become increasingly naturalized (and, importantly, mandated by institutions) as the parameters through which the possibilities of course design and learning are imagined. In their web-based monograph on writing workflows, Tim Lockridge and Derek Van Ittersum write that "for many academics, knowledge work is grounded in…[e]nterprise writing software, which has historically been Microsoft Word and is now increasingly Google Docs."[107] Institutional defaults become intellectual defaults as "Word, for many, seems like an appropriate place to begin and end a writing task." Yet, when personal computers were an emerging technology—at home or at school—researchers examined the word processing program's effect on writing instruction compared to handwritten work with "ambiguous" findings in terms of the effects on quality of writing, attitudes towards writing, and engagement in the writing and revision process. More unambiguously, however, word processors freed writers from the manual effort of composing and revising, providing a "fluid" composing environment that more cleanly displayed the current state of a text (albeit by visually suppressing the versioning that would be more clearly marked

[106] There are, of course, important disciplinary platform conventions that fall beyond the generalizability of word processing programs similar to MS Word, e.g., LaTeX and Markdown in STEM fields.
[107] Indeed, this dissertation was drafted exclusively in Google Docs with backup copies periodically downloaded as Microsoft Word docx files.

on paper) and freeing the writer to focus on high order concerns and compose greater quantities in shorter amounts of time (Bangert-Drowns 86-88). Critical responses to emerging technologies (in this case, the word processor) are particularly telling for the questions that they articulate more explicitly and self-consciously than in later work that takes for granted how technologies operate. In the case of programs that facilitate writing, to what degree do writers take advantage of the increased ease of iterability and versioning that leads to a more reflective and nuanced writing practice? Conversely, what limitations do programs place on the writing process and on how we may perceive and encounter our writing?[108]

Now a standard approach for research and teaching in composition and growing out of the computers and writing community, multimodal composition has answered the call that these questions represent in terms of looking to a wide range of compositional modalities and genres in which students meaningfully reflect on process and product and encounter the rhetorical situation in authentic ways. Where students would have written a series of alphabetic texts they now might compose video essays, podcasts, visual or graphic essays, mixed media collections or portfolios, websites, material or digital objects, games, exhibits, and performances. As Cynthia Selfe argues, multimodal composing encourages students to use all "valuable semiotic resources for making meaning" including but not limited to alphabetic print (617). Her apologia for multimodality and aural composing establishes that what is at stake is no less than students' "rhetorical sovereignty" —their "rights and responsibilities...to identify their own communicative needs and to represent their own identities, to select the right tools for the communicative contexts within which they

[108] Perhaps one of the most common examples of a (contested) limitation in this sense is the word processor's default visual interface based on physical sheets of paper of a particular size (an interesting structural or UX-based instance of the typically iconographic skeuomorph). Some researchers, however, have viewed the "foreground[ing]" of visual presentation on the printed page "as an [additional] area of rhetorical in(ter)vention" (Takayoshi 246).

operate, and to think critically and carefully about the meaning that they and others compose" (618). In other words, theories of multimodality address "semiotic codes" rather than technical mastery of tools or platforms per se (Kress and van Leeuwen 177). A course on composing for the web, for instance, would include but aim beyond learning outcomes regarding technical mastery of content management systems such as Wordpress and coding languages such as JavaScript; rather, the overall goals would involve rhetorical and semiotic analysis of how websites and web-authored publications convey meaning to and exert agency on readers.

The prevalence of digital tools in multimodal composition scholarship and practice have led to associations with digital humanities pedagogy in the scholarly imagination. In their twenty-year retrospective on computers and writing, Jennifer Marlow and James P. Purdy note that in 2009 *digital humanities* began to appear in the field's organization and publication of record—the Computers and Writing conference and *Computers and Composition*—coinciding with the digital humanities moment that I have explored in the previous chapter.[109] However, they find that the relationship is not reciprocated; while computers and writing scholars take up the question of the field's relationship with digital humanities (one marked with alignment, tension, and separation), publications in the digital humanities often remain silent on the matter of computers and writing. One of the exceptions that Marlow and Purdy note is the 2012 edited collection *Digital Humanities Pedagogy*; its chapter focusing on first-year writing, for example, draws significantly from multimodal literacy frameworks advanced by Cynthia Selfe, Anne

[109] Marlow and Purdy speculate that the prevalence of "digital humanities" in the conference program was tied to the amenability of the conference theme; the year-to-year frequency of the term oscillates dramatically from 2009 to 2015 in their data. They frame their digital book *Are We There Yet?* as a celebration of and the intellectual sequel to Gail E. Hawisher, Cynthia L. Selfe, Paul LeBlanc, and Charles Moran's *Computers and the Teaching of Writing in American Higher Education, 1979-1994: A History*, Ablex Publishing Corporation, 1996.

Francis Wysocki, and Johndan Johnson-Eilola (Bjork 100).[110] I offer this dissertation as well, and this chapter in particular, as an effort to reckon seriously with computers and writing (and composition studies broadly conceived) to advance pedagogy in the digital humanities, which Marlow and Purdy identify as one of the three "site[s] of connection" between the scholarly areas along with the maker movement and digital rhetoric. J. Elizabeth Clark synthesizes these shared concerns in a clarion call following Kathleen Blake Yancey's notion of writing in the 21st century,[111] merging the insights of multimodal composing and digital literacies to form a "digital imperative" that she presents as the defining aspiration of "21st-century pedagogy." What matters for this imperative, she argues, is not that a student master a particular technical or digital platform (these will always be in "flux," she explains) but rather that we "challeng[e] traditional notions of essayistic literacy" in curricula and pedagogical approaches (27-28, 34).

In addition to essayistic *literacy* I would add essayistic *workflow* as typically enacted for writing-based tasks and assignments in educational contexts. The digital imperative, in other words, implicates not only an expansion of our understanding of semiotic systems but also an adaptability in how we imagine and go about the act of composing itself in relation to the affordances and constraints of particular tools and platforms. Lockridge and Van Ittersum invoke this notion of workflow "as a lens to examine the often omitted tools, material conditions, and activities of writing," which "highlights the importance of writing tools and allows us to consider how tools shape activity and, in turn, how activity shapes tools." They are careful to note that

[110] A second collection of note is *Digital Pedagogy in the Humanities: Concepts, Models, and Experiments*, edited by Rebecca Frost Davis, Matthew K. Gold, Katherina D. Harris, and Jentery Sayers, Modern Language Association, 2020, digitalpedagogy.hcommons.org/. While it does not explicitly evoke computers and writing as a disciplinary precedent, Marlow and Purdy note that it draws from C&W scholars and explores concepts central to the field.
[111] Kathleen Blake Yancey, "Writing in the 21st Century: A Report from the National Council of Teachers of English," National Council of Teachers of English (NCTE), 2009, cdn.ncte.org/nctefiles/press/yancey_final.pdf.

any attempt to recenter tools in academic and especially humanistic discourse should heed the "vagueness about the term" that can allow *workflow* to evoke "contemporary capitalism and the business push to replicate tasks, to distribute tasks among workers, or to industrialize a process so that manufacturing can proceed without the expertise of a single worker." Against this grain, Lockridge and Van Ittersum draw from qualitative studies of Markdown influencers to redefine workflow "as a personal process, rubric, and metacognitive lens" that provides "a means of evaluating the components, processes, procedures, and technologies" of writers' work. Through this lens writers "can look at their broader writing process and begin to analyze the connections, intersections, and fissures within the component parts of their work. And it is a lens that is fully intertwined in writing technologies." The pedagogical project behind English 1110.01 "Codes" adds to the expansion of essayistic literacies in composition studies within an *alphabetic* textual modality with the inclusion of metacognitively-oriented markup language, and it also adds to the expansion of essayistic workflows with the use of a code editor, class corpus file, subversion repository, and browser-based website. Defamiliarizing, atomizing, and collectivizing the writing process in an otherwise second-nature alphabetic modality,[112] the workflows in English 1110.01 "Codes" sought to reinforce the metacognitive thrust of the markup with granularity, iteration, transparency, accumulation, and manipulability.

The most apparent difference in writing workflow for students in the course was the use of the <oXygen/> code editor instead of a standard word processor. A few students in the course

[112] In this way my work also follows the lead of technical and professional writing scholars who for years have drawn our attention to the range of genres, styles, purposes, platforms, and materialities of (digital) alphabetic texts *in situ* (Blythe, Lauer, and Curran 273-74). Similarly, in a 1998 computers and writing conference town hall, Gail Hawisher pronounced in a position statement that "[w]riting is headed into *more* of everything (alphabetic text, multi-media texts, hypertexts)" not only with respect to volume (e.g., length, frequency) but also with respect to type (e.g., genre, medium, software).

had experience with code editors but had used them only for engineering or computer science. Figure 10 below shows a screenshot of the <oXygen/> interface as it displays a portion of the instructor-generated example of the primary source analysis.

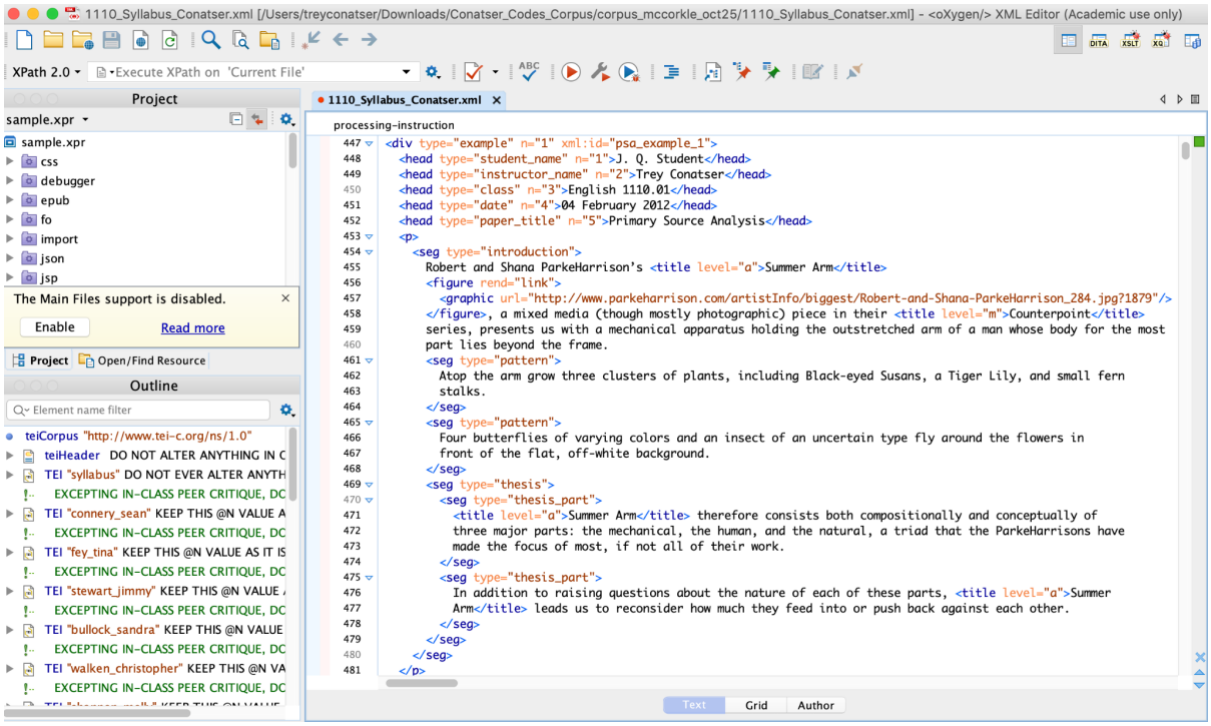


Figure 10: screenshot of the <oXygen/> interface showing an excerpt of the example primary source analysis

While the text editing window functions similarly to any word processor or text entry interface in the left-to-right and top-to-bottom format, writers notice differences almost immediately such as the lack of a defined page-area that forces line breaks (i.e., a line can proceed infinitely to the right), the nested structure of elements with forced indents, the color coding of the markup, and the unreadable spaces to the left of the beginning of a line (i.e., machine-readable blank spaces may occur only *after* a character or element begins a line). The text styling options prominently displayed at the top of a word processor are entirely absent, as are any automated formatting and correcting functions. <oXygen/> includes a spell-check feature that underlines words in red,

while badly formed markup (e.g., an overlapping hierarchy, an element that is not closed) is also highlighted in the editing window and noted with colored notches along the vertical scroll bar where the issues occur in the code file. Once the corpus file began to expand beyond a few thousand lines long, these notches were critical for efficient troubleshooting given that the XML file would fail to compile for a transformation unless all markup was well formed. In theory, though perhaps not so much in practice, the notches invited students to help each other if they noticed them in a peer's section of the corpus file.

The corpus file represented the most radical change for students' writing workflows. Rather than using only their own machines' storage or only cloud-based storage, students worked in both modes at once, writing all assignments from daily activities to major assessments on a single XML file that, by the end of the semester, exceeded 42000 lines in length. In addition to the different mechanics for saving work and accessing the most recent version of the corpus file, students wrote in a context of collective file ownership given that even while they worked locally on their own machines they still were editing a record of all 25 students' coursework. Within the `<teiCorpus>` root element and after the initial `<teiHeader>` for the corpus file (which contained the syllabus, assignment instructions and examples, and other information), each student worked within a `<TEI>` sibling element that contained their writing and other assignments for the course. While `<TEI>` elements typically signify individual documents within a corpus, they served as dividers between students' bodies of work. Each `<TEI>` element contained its own `<teiHeader>` for metadata about the student. I built these sections prior to the first day of class so that for each activity or assignment students needed only to find the appropriate section of their `<TEI>` element and begin to write. More advanced courses with different learning outcomes may benefit from

asking students to determine the overall structure of the corpus file but for first-year writing it was important that students were able to focus on acclimating to the markup that would develop their meta-awareness about composition. Figure 11 below shows a simplified structure for both the class corpus file and an individual student's `<TEI>` element.

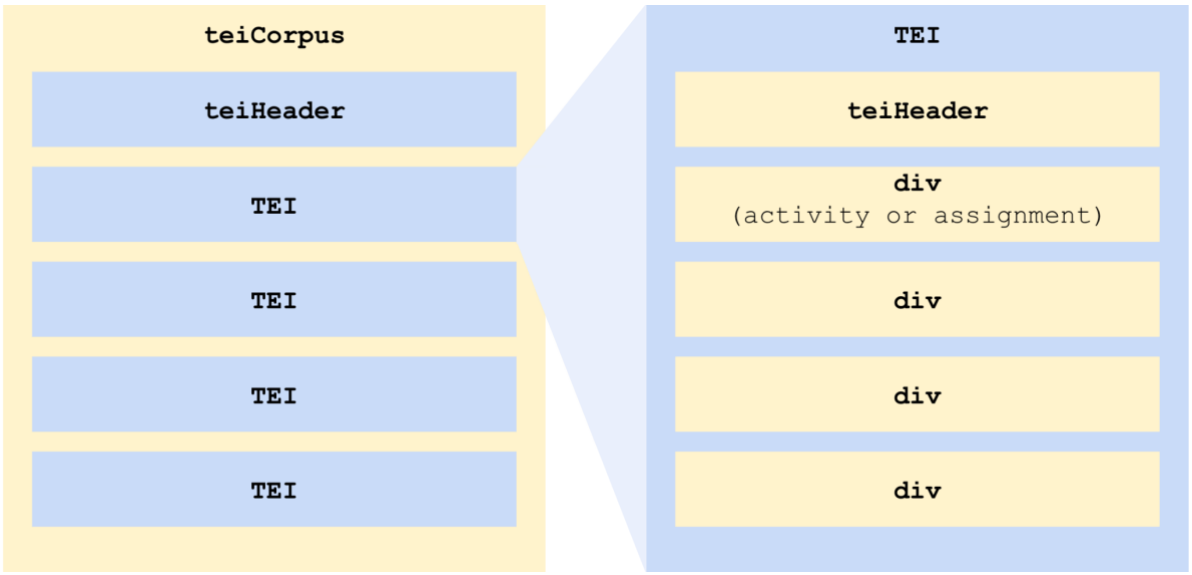


Figure 11: simplified structure of the class corpus file and individual `<TEI>` element

Given that students worked on a shared file, the process of saving and reconciling their changes was perhaps the most unfamiliar aspect of working with the corpus file. With a program such as Microsoft Word, the writer engages the 'save' function at whatever frequency they please (some with only once at the end of a writing session, others with compulsive saves after every minor change or addition) and alters the file locally on their device's storage. Cloud-based word processors such as Google Docs automate the save function to follow any changes made by any number of writers on the same document; several writers, for example, may be editing the same paragraph at the same time while Docs saves all changes at a granular level without need for any user intervention. Word, Docs, and similar platforms keep a version history of the file

showing the sequence of discrete changes between saves with metadata such as timestamps and allowing the writer to revert to previous versions of the file. In contrast, composing in <oXygen/> made the process of updating, saving, and reconciling changes much more explicit for student writers. <oXygen/>'s built-in Subversion (SVN) client allows users to connect to an online repository, 'check out' the current version of a code file, make changes locally on their device, integrate any updates in their local file that may have been committed to the repository file after they first checked it out, and 'commit' their changes to the repository as the most current, updated version of the corpus file. Figure 12 below offers a visual representation of the subversion process.

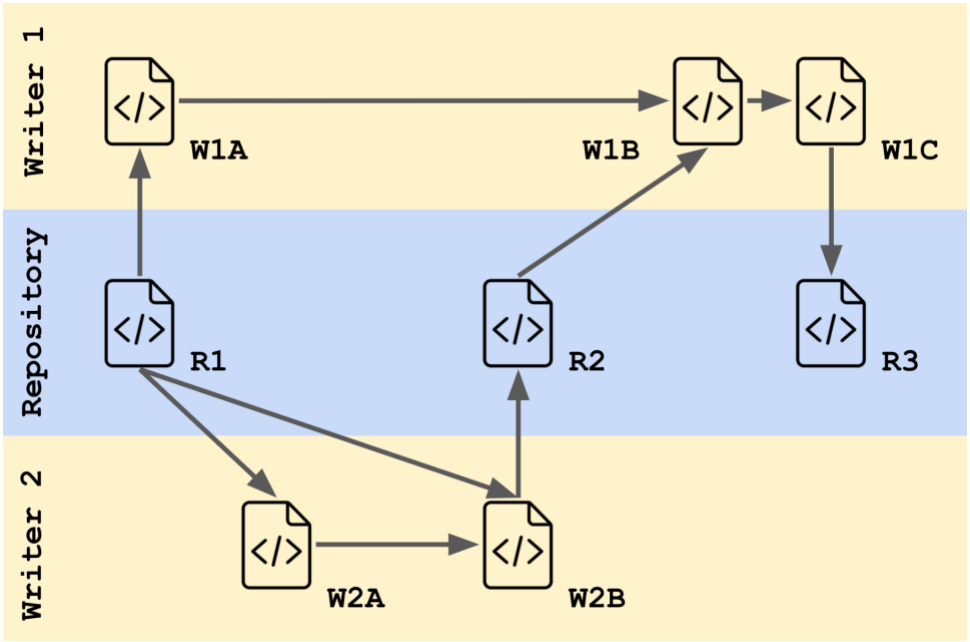


Figure 12: Subversion repository process with two writers (icon "coding" by Adrian Adam from the Noun Project)

In this illustration, Writer 1 checks out the current version of the repository file R1 and begins to make changes to the file W1A on their local device. During this time, Writer 2 checks out the current version of the repository file, which is still R1, and begins to make changes to the file

W2A on their local device. Writer 2 finishes their changes first and checks for other changes that have been made to the repository file since they created W2A on their local device. Because the repository file is still R1 (i.e., no changes have been made to it yet), Writer 2's local file W2B is not altered beyond the changes they made locally. Writer 2 then commits the file W2B as the updated version of the repository file R2. Later, Writer 1 saves their changes on their local device as file W1B and checks for other changes that have been made to the repository file. Finding that changes have been made, the subversion client merges them with W1B on Writer 1's local device to create file W1C, which includes changes made by both Writer 1 and Writer 2. Writer 1 then commits the file W1C as the updated version of the repository file R3. In this way the subversion repository creates a growing 'stack' of file versions, one on top of the other. The version history shows changes from version to version as it would in any other program, but with each commit writers are also prompted to describe/summarize the changes they've made to the repository file. With authorial attribution, this description accompanies the new file as an added layer of human-generated, descriptive metadata.[113]

The process outlined in figure 12 is simplified to two writers for explanatory purposes; in the class itself 25 people committed changes to the corpus file. Due to the deadline-driven nature of coursework, students checked out, modified, and committed changes to the corpus file within concentrated periods of time prior to a due date. So long as students followed the subversion process properly and left no lag time between checking for updates and committing changes (i.e., integrating changes into their local file that other students had made since they last checked out a

[113] There are other models for cloud-based version control workflows that employ multiple 'branches' of development that grow from (and merge back into) a central 'trunk' repository. For the purposes of this class and given the constraints under which students worked, a single stack of versions of a single file was more than enough to manage our work effectively.

version of the corpus file) there were no conflicts in the repository even during high-traffic commit periods. However, if students worked on an older version of the corpus file, forgot to save their changes locally before pulling recent updates from the repository, or failed to check for updates moments before committing changes, they would either erase other students' work without raising any flags (because the subversion client would read the erasure as a legitimate change in the corpus file) or trigger a conflict in the commit process whereby two irreconcilable versions of the corpus file were in competition. The latter scenario occurs when editors change the same lines of code at roughly the same time. Because students typically worked in their own `<TEI>` elements chances of a conflict were low except for times when the class was engaging in peer review (which involved students inserting `<note>` elements in other students' markup) or in-class activities that required an additional `<TEI>` element be added to the corpus file wherein all students worked for the duration of the class meeting. Still, non-conflict-based erasures were sometimes common. Badly-formed markup (e.g., elements left open, overlapping hierarchies) or invalid markup (e.g., elements or attributes not within the TEI Guidelines due to typos or other errors) that had been fixed by one student in a previous version of the corpus file suddenly would return in a newer version, or edits that one student had made to their assignment would revert to a previous draft (or text would be erased altogether). Most problems were easily fixed because the commit log provided access to all previous versions of the corpus file; a writer needed only to locate their own most recent commit, copy their `<TEI>` element from that version of the corpus file into the most recent version, and re-commit the latter version to the repository. On one occasion early in the semester so many students committed problematic versions of the corpus file to the repository during the evening before a deadline that we needed to take the next class

meeting to pore over the commit log and restore each student's work to its most recent form. I framed the occasion as a learning moment and an opportunity to come together as a class to reaffirm students' collective responsibility to each other's learning and success.

The subversion process allowed all students to work on the 'same' shared corpus file that provided a radical transparency in the form of access to the most recently committed version of all writers' work at any time during the term. While students were 'seeing the code' in terms of how the markup illuminated their rhetorical and writerly choices, they also were seeing *each other's* code—and writing—as concurrently evolving examples of both the assignment at hand and students' overall thinking about their artifacts and arguments. Typically, students encounter other students' writing a limited number of times during a writing course, usually during peer review activities. Scholarship on peer review most often focuses on the process and practices of giving and receiving feedback, as well as ways of facilitating peer review sessions with respect to either scaffolding and facilitation (i.e., pedagogy) or technical platforms (e.g., for online and large-enrollment courses).[114] As a byproduct of the peer review process, however, students come to understand their peers' intellectual journeys as well as their current challenges, opportunities, and goals; in short, we cannot understate the value of students *being exposed to* peers' work and ideas in a systematic and intentional way. Expert advice on teaching with peer review activities sometimes gestures to the benefits of reading peers' work, while a commonplace among advice on writing pedagogy involves the use of authentic and/or student-generated examples as opposed to, say, relying exclusively on professionally published pieces to guide students' development as

[114] For a brief review of the literature on peer review up to 2010, see Elizabth A. Flynn, "Re-Viewing Peer Review," *The Writing Instructor*, 2011, files.eric.ed.gov/fulltext/EJ959705.pdf.

writers. In general, peer-driven learning seeks to overcome what psychologist Steven Pinker describes as "the curse of knowledge": the expert tendency to think and communicate about an area of expertise in ways that become increasingly inaccessible and illegible for novices (74-75). Physicist Eric Mazur—a longtime evangelizer of peer instruction and active learning in STEM education—draws a line between Pinker's curse of knowledge, originally applied in the context of technical and scholarly communication, and the challenges that instructors face in terms of meeting students where they are as novice learners struggling with concepts and practices that have become second nature to experts in the field. Peer instruction, Mazur emphasizes, provides a significant way for students to learn from others who are just beginning to understand or master concepts, and I would add that it also exposes learners to their peers' thinking in ways similar to peer review (and, ultimately, similar to working on a shared corpus file).

Peer review itself looked different in the corpus file, where feedback and comments that typically are spoken in conversation or dashed off on individual drafts (printed or digital) were typed at the most relevant location directly in the middle of the assignment text using the `<note>` element with a `@type` attribute distinguishing it from other `<note>` elements in the file (e.g., for annotated bibliography entries) and a `@resp` or responsibility attribute indicating the author:

```
<p>
    Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor
    incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud
    exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.
        <note type="feedback" resp="studentname">
            Peer feedback here.
        </note>
    Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt
    mollit anim id est laborum.
</p>
```

In addition to becoming a permanent part of the markup and writing in a student's assignment in the corpus file, feedback notes also showed in the HTML views of student work with the help of XSL code that transformed `<note>` elements into (a) superscript numbers that displayed the full text of the note as a tooltip popup if the mouse pointer hovered over it; and (b) traditional end notes that followed the main body of the text. 'Interrupting' the main text of an assignment with `<note>` elements in the XML file does not pose a problem for XSL transformation because the parsed character data within any elements may be suppressed—i.e., overlooked or not printed with the rest of the text—by including an 'empty' template matching the desired element, in this case `<xsl:template match="note"/>`. The slash 'closes' the template without performing other actions, which instructs the transformation to do nothing, i.e., ignore the element as well as what it contains, when encountering `<note>` elements any time `<xsl:apply-templates/>` is called. This does not prevent other actions from printing the content of `<note>` elements, however, using the `<xsl:for-each>` loop to generate endnotes with the `<xsl:value-of>` command. What emerges in the XML file itself is an accumulation of reader responses to students' writing in addition to the writing itself; both peer and instructor comments become part of the text and, interestingly, share the same hierarchical level in the structure of the XML. Figure 13 below shows a tooltip popup in the HTML view, which displays the content of a `<note>` element when the mouse hovers over the superscript number.

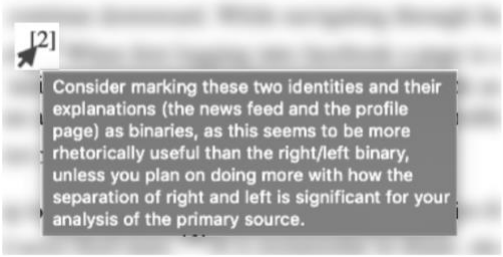


Figure 13: instructor comment on student work (blurred) displayed as a tooltip popup in the HTML view

Especially with the exponential growth of digitally based teaching and learning activities in the wake of COVID-19, concerns over student privacy and data ownership have likewise grown with the development of analytics-driven learning management systems, numerous third-party plugins and platforms, and surveillance technologies such as online test proctoring. While peer review has always asked students to provide feedback on each other's work, *instructor* comments have typically remained a private matter given their direct relationship to a student's grade. FERPA, of course, is the legal basis for this obligation of privacy regarding a student's education records, but it also is part of the social contract of a learning community that students feel protected from intrusion into their privacy and that sensitive information remains theirs to do with as they will. Accordingly, instructor comments were aware of a larger audience: not only the student-author but also their peers; these comments, then, needed to be instructive for all students in terms of modeling an expert's response to a novice's work similar to feedback given at public assessment fora such as the critique in art education. Students were told and reminded that public comments were not evaluative in nature, but rather provocative in terms of asking questions and proposing ideas in the spirit of formative assessment. More explicitly, they were told, instructor comments had no relation (via tone and tenor, word count, frequency, etc.) to the grades earned. Evaluative feedback and grades were provided in the FERPA-compliant learning management system. Corpus file, subversion repository, peer and instructor comments: the move from word processor to code editor involved more than a change in writing workflow per se; it sought to make visible writers' mental moves—for themselves as well as their peers—fostering a more robust learning community.

<Assignments, Markup, and Transformations/>

Assignment markup schemes reinforced students' meta-awareness about composition in a number of ways, from identifying their own rhetorical moves as well as rhetorical and analytical components of their primary sources to explicitly tracing lines of inquiry across assignments. The markup scheme for each assignment was idiosyncratic and designed by the instructor based on (a) the particular goals of the assignment in the context of the first-year writing course, (b) the general goals of writing in the genre and rhetorical situation that the assignment represents, and (c) issues that students typically encounter when practicing academic and professional writing. Appendix A contains the markup scheme and a fully marked-up example of the primary source analysis, but two aspects of the markup scheme merit attention here in the context of developing meta-awareness and cuing students to one of the more universal and challenging goals of writing courses and assignments: the strategic use of evidence-based reasoning broadly construed as well as adapted to the genre, purpose, and audience (in this case, interpretive and speculative analysis of a primary source without research expectations), as well as the articulation of that reasoning in ways that clearly and persuasively describe the evidence itself and the writer's interpretation of it for the reader. The latter goal was particularly important for me given anecdotal observations that while students perceive the importance of evidence-based reasoning, they often invoke evidence in ways that are unclear or without context, and they often do not move beyond the citation of evidence itself—i.e., assuming that the *meaning* of the evidence is self-evident and, even further, *persuasive* for the reader—to offer an interpretation of that evidence in the context

of the overall argument or goals of their writing. Two required markup elements of the primary source analysis prompted students to develop meta-awareness of this practice:

```
<seg type="ev_interp">
    <seg type="evidence">evidence (i.e., observable details) from the primary source</seg>
    <seg type="interpretation">interpretation based on the corresponding evidence</seg>
</seg>
<!-- At least one time (though I strongly encourage you to do more), match evidence from the
primary source with the interpretations you draw from it. -->

<seg type="anomaly">
    <seg type="anomaly_ps">your observation of an anomaly in the primary source</seg>
    <seg type="anomaly_sig">your explanation or speculation as to the significance</seg>
</seg>
<!-- Tag at least one moment when you discuss an anomaly in the primary source as well as your
explanation of the significance of that anomaly.-->
```

While the templates appear to lock writers into a particular sequence in which the evidence itself always precedes a discussion of its significance and meaning, the way in which XPath navigates XML nodes for XSL transformations allows for sibling elements to appear in any order. By way of this technical affordance students understood that interpretation may precede or proceed from the evidence, depending on the needs at a particular moment in the writing. The example primary source analysis, taking as its subject Robert and Shana ParkeHarrison's *Summer Arm* (Figure 14 below), applied the above markup in the following ways.

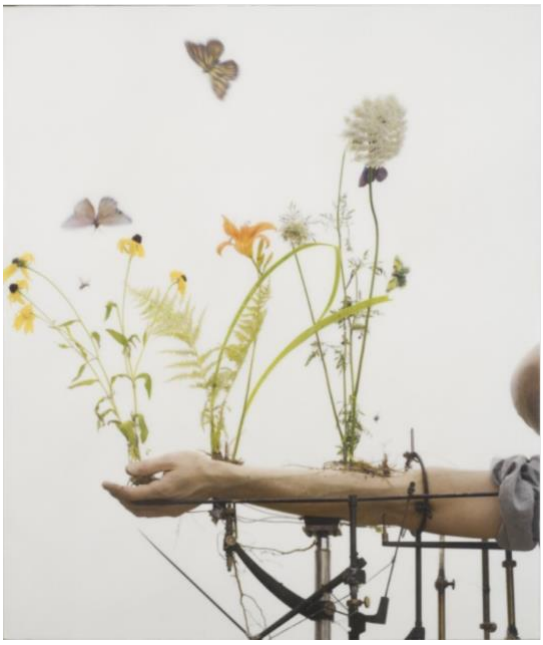


Figure 14: Robert and Shana ParkeHarrison's *Summer Arm*

```
<seg type="ev_interp">
    <seg type="interpretation">
        Moreover, the device both threatens and supports the man's arm.
    </seg>
    <seg type="evidence">
        Rods taper to needle-like points uncomfortably close to the man's flesh, and a circular
        component just above the elbow seems to function as a clamp holding the arm in place: a
        buttress, or, conversely, a restraint.
    </seg>
</seg>

<seg type="anomaly">
    <seg type="anomaly_ps">
        Though we see only the very top of his head, it clearly tilts deeply in the direction
        of the extended arm,
    </seg>
    <seg type="anomaly_sig">
        perhaps compensating for the awkward hyperextension, or perhaps resting out of sheer
        exhaustion. Indeed, we don't know how long the man has been in the apparatus; if the
        plants actually have grown on his arm, he may very well have been in this position for
        quite a while. This character also appears with arm-sprouted plants at other stages of
        the growth cycle in <title level="a">Winter Arm</title> and honeycombs tied to his arm
        in <title level="a">Spring Arm</title> in the <title level="m">Counterpoint Gray
        Dawn</title> series, though he is not bound by an apparatus in either images. If there
        is an implied story, it is mythic and unclear, the man appearing as an archetype across
        the disjointed mises en scène.
    </seg>
</seg>
```

In addition to meta-awareness at particular moments of composition, the markup allowed students to track the development of their larger inquiries across the semester. First-year writing students may compartmentalize scaffolded assignments to the detriment of a design-oriented approach to their projects—i.e, they might consider the assignment in isolation from the guiding logic and possibilities of their larger project goals and research questions. A goal of first-year writing often involves familiarizing students with the intellectual journey of a project that spans multiple drafts, genres, and modalities. Indeed, the *process* of writing-based inquiry and communication is one of the four major categories in the WPA Outcomes Statement. Yet, college students—particularly those whose learning continues to be disrupted and altered by a global pandemic—often struggle with the executive functions that would otherwise allow them to track the development of a larger and multi-pronged project. CAST, the international organization that publishes the Universal Design for Learning (UDL) Guidelines, defines executive functioning as the ability "to overcome impulsive, short-term reactions to their environment and instead to set long-term goals, plan effective strategies for reaching those goals, monitor [one's] progress, and modify strategies as needed." Put differently, executive functions involve overcoming a present-oriented survival mindset to a past-and future-oriented reflective and managerial mindset. In the first-year writing class, long-term planning and monitoring the development of a writing-based project may quickly drop off as students struggle to balance their coursework, extracurriculars, jobs, personal responsibilities, acute challenges, and chronic hardships. In particular, students' goals and research questions should guide their work through the semester. The primary source analysis, as the first assignment in the scaffolded sequence, required that students conclude with compelling and effective research questions to which their

observations and speculations had led them. The questions would inform students' research efforts, selection of secondary sources, and synthesis of insights from those sources into their fully developed arguments. An attribute in the markup for the second assignment, the annotated bibliography, sought to reinforce this aspect of students' project management by asking them to match selected secondary sources with research questions:

```
<bibl type="type_here" xml:id="id_here" n="alphabetical_organizer" corresp="xml:id(s)_of_
corresponding_research_question(s)">
    <!-- MLA works cited entry here -->
    <note type="ab_annotation">
        <!-- Annotation here -->
    </note>
</bibl>
```

The `@corresp` or corresponds attribute in the `<bibl>` or bibliographic citation element linked the bibliography entry to the research questions tagged in primary source analysis via their unique identifiers encoded as `@xml:id` attributes:

```
<seg type="research_question" xml:id="surname_rq_#">research question</seg>
```

In the main `<note>` entry for the secondary source, students also tagged `<seg>` elements with attribute types for the occupation of the author(s), the main argument(s) of the source, and the relevance to their overall project. The HTML view presented a traditional view of the annotated bibliography with full annotations in addition to a more granular breakdown drawing from what students tagged; this latter 'analysis' view (as it was called on the class corpus website) called attention to what the student had tagged. Figure 15 below shows a screenshot of the HTML view with an instructor-generated annotated bibliography entry (also drawing from D. Travers Scott's review of Steve Jones's *Against Technology* in the *International Journal of Communication*).

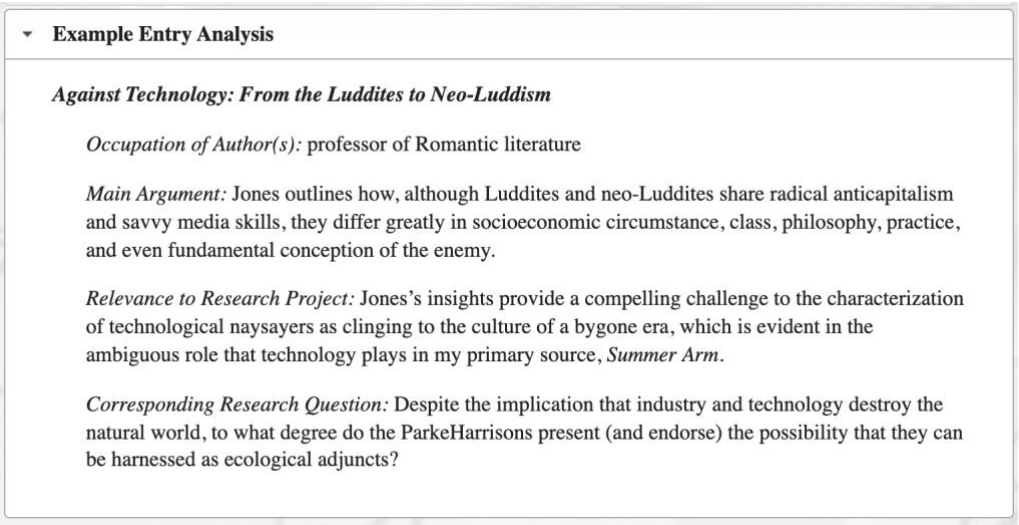


Figure 15: HTML view of example annotation breakdown featuring corresponding research question (with language from D. Travers Scott's review of Steve Jones's *Against Technology* in the *International Journal of Communication*, vol. 1, 2007, ijoc.org/index.php/ijoc/article/download/21/1).

The markup in this case asked the student to add a layer of information regarding their thinking about how each source furthered their inquiry as it was established by their research questions. It also explicitly cued this thinking in the first place where students may engage in a less strategic, looser process of identifying and selecting sources. The markup scheme additionally sought to demystify the annotated bibliography entry, making the expectations transparent, purpose-based, and specific to foster a rhetorically savvy approach to summary-based writing.

As the students moved from the annotated bibliography to later assignments, they worked on integrating ideas, information, and other elements from their sources into their writing. This is no simple task and represents perhaps one of the most challenging aspects of first- and second-year writing courses if not all research- and/or persuasion-oriented writing courses and activities in an undergraduate program. Instructors are familiar with the 'fly-by' approach to citing sources where claims or information appear without much context in student writing, 'dropped' into the

text as if their presence itself makes their *function* in the student's writing self-evident as well as persuasive for the reader. The imperative to see the code here aligns with the need for a reader (instructor or otherwise) to *see the thinking* in student writing, particularly when already complex arguments attempt to triangulate evidence, ideas, questions, claims, themes, and other important information in ways that are targeted to the needs of the moment. Part of this writerly imperative, for students, involves cultivating an awareness of the myriad purposes for which a writer invokes other voices and sources beyond offering ostensibly clear 'proof' that their claims are correct. The markup scheme of the secondary source integration—an enhancement of the primary source analysis with findings from sources included in the annotated bibliography—asked students to use a nested, three-part tag structure when citing sources:

```
<seg type="integrating_sources">
    <seg type="me">
        your initial insight about the primary source
        <!-- i.e., something you observe, interpret, infer, etc., in analyzing the source -->
    </seg>
    <seg type="source">
        citation of related insight(s) from a secondary source
        <!-- i.e., the in-text citation: quoted, paraphrased, or both, with parenthetical -->
    </seg>
    <seg type="synthesis">
        your new, clarified, qualified, or modified insight(s)
        <!-- i.e., how the cited information, ideas, etc., inform your thinking about both the
        point at hand as well as your overall argument, if appropriate →
        <!-- e.g., so what? what does it mean? what are you thinking *now*? -->
    </seg>
</seg>
```

The tripartite structure may not serve all instances of citation, and students were told not to force their thinking into a particular mold if that mold became a barrier to clear and persuasive writing. So long as *some* citations were marked up as such, students successfully met the requirement. An additional component to the markup could have asked students to identify the writerly move they

were making with the source citation using an `@subtype` attribute for the container `<seg>` element with a controlled vocabulary for the attribute values, e.g., context, example, contrast, validation. Through all assignments, the markup sought to make visible and conspicuous—and thus invite a slow and critical reflection on—rhetorical and writerly practices at the heart of first-year writing classes as well as composition and communication at large. TEI compliant XML, as a particular type of markup language (in turn as a particular type of code), cultivated a meta-awareness about composition and made visible to writers as well as readers the mental moves and choices that are foundational to an effective writing practice for college coursework and beyond.

<Responses, Limitations, and Possibilities/>

A small number of scholars and students offered responses to the course after a HASTAC blog post about it was reworked for publication in the open-access *Journal of Digital Humanities* shortly after the end of the semester.[115] While they acknowledge the unique nature of the course as well as the precedent it has set for expanding coding practices and literacies into the writing curriculum, they also ask how a different approach could have traded less in functional literacies and instructor-centered teaching. In her doctoral dissertation, *Critical Experiential Learning and Rhetorical Interventions in New Media Ecologies*, Jennifer Niester-Mika briefly addresses the XML first-year writing class while articulating her vision for critical literacies and new media in

[115] The article had generated a handful of student response blog posts for at least two courses that had assigned it as reading, but those websites have since been taken down. Here I cite two doctoral dissertations that respond to the course design in the context of digital pedagogy and rhetorical code studies. See "Changing Medium, Transforming Composition," HASTAC Scholars, 18 April 2013, www.hastac.org/blogs/conatser4/2013/04/18/changing-medium-transforming-composition; "Changing Medium, Transforming Composition," *Journal of Digital Humanities* vol. 2, no. 2, 2013, journalofdigitalhumanities.org/2-2/changing-medium-transforming-composition-by-trey-conatser/.

the humanities curriculum. Against an ideal critical literacy that attends to "tools and knowledge that writers/performers/artists can draw from in the emerging moment of production" and reinvests in "a civic foundation [in] what has become a largely post-critical field," Niester-Mika places rhetorical and functional literacies as less ambitious and less effective goals for courses that engage students in digital production and knowledge-making, privileging "technique/skill" to the detriment of "critical theory" understood vis-à-vis Giroux, et. al (102). "Functional literacy is best demonstrated by new efforts to insert coding into composition classrooms," she contends, citing my course as an example. "[Becoming] more aware of...writing choices" and "enhanc[ing] the writing process," in Niester-Mika's thinking, fall short as learning outcomes when compared to the promise of "how student[s] can use coding to refashion their worlds" (101). The critique is a fair one insofar as the constraints of the *Journal of Digital Humanities* publication required that I neglect theoretical orientations in favor of a functional focus on the course format and markup, and also insofar as the 'set' nature for English 1110.01—with my own positionality as a graduate instructor—required that I graft the markup process on top of a curriculum without changing it at all (i.e., cleaving precisely to learning outcomes regarding writing choices and processes). The type of activity that Niester-Mika offers as a more ideal counterexample, however, errs in favor of depth over breadth; the "cultural hacking," as she calls it, draws from culture jamming, remix activities, and found-art traditions to ask students to "compose an argumentative video essay… using found imagery/video/audio and remixing it" (105-07). While Niester-Mika asks how new media courses with learning goals grounded in critical and civic interventions can fully live up to the promise of the theories that inspire them, I engage in a broader inquiry regarding the uses of code for humanities and liberal arts education. What are the ways in which code may enhance

and transform learning in a variety of curricular environments, from the more pointed disruptions that Niester-Mika hopes to cultivate, to the more diffuse effects of using code such as markup language as a means of alphabetic composition? Through more oblique processes, students may begin to engage in the kind of critical thinking and meta-awareness that work constructively with more explicitly interventionist aims. Regardless, I have attempted in this dissertation to construct a stronger theoretical and critical foundation for my case studies and model a scholarly approach to teaching that resists instrumentalism.

A second response to the course focuses on students' intellectual agency, the hierarchy of learning outcomes, and the positionality of the instructor. Kevin G. Smith's doctoral dissertation *(Re)making/(Re)marking: Genre and Markup in the Writing Classroom* sets out a comprehensive vision and method for engaging students in writing with markup using rhetorical genre studies as a design framework and site for intervention. Indeed, "how [writing] students make and make sense of genre knowledge" when they "interfac[e] with it through XML" is the central concern of Smith's project (3). Understanding genre as a site of change and rhetorical invention as well as the more familiar concept of recognizable parameters and moves that 'make' a particular type of writing, Smith explores the "productive possibilities for representing the dynamics of rhetorical genres through explicit modeling" in XML (11-13). He cites my first-year writing class among "a small body of work that considers the rhetorical capacities of markup," which he distinguishes from the more widespread use of text encoding in digital humanities work to produce digital editions or databases of print- or manuscript-based texts. Using markup to *compose* texts rather than represent them, he summarizes, places "an emphasis on the process, rather than reception of encoded texts" (9-10). While the next chapter seeks to complicate this

distinction with a sense of how digital scholarly editing and text encoding engages students in process-based thinking and the production of (new, transformed) texts, the case study in this chapter draws Smith's attention for the means by which the markup schemes were determined and applied in class. Speaking of both my markup-based writing course and <emma>, a project at the University of Georgia,[116] Smith finds that "markup functioned as a top-down system for authoring texts" whereby, in my case, the instructor determined *a priori* the tags that would constitute the markup scheme for each assignment (29). In other words, students were given a schema and asked to apply it to their work, as opposed to a more bottom-up approach that would ask students to model the text at hand and collectively (and iteratively) determine the markup that best represents it. Smith's own course design drew from Wendell Pietz's notion of exploratory markup as well as "inductive methods of coding textual data" in the social sciences to engage students in "emergent model[s]" of their writing as instances of larger rhetorical genres. (75-76).[117] Smith reports that "a model of a particular genre 'emerged' as the more exploratory codes cultivated by students were consolidated and codified in the collaborative schema files" that students then applied to their own writing as markup (76). Figures 16 and 17

[116] See Christy Desmet, et al., "<emma>: Re-forming Composition with XML," *Literary and Linguistic Computing*, vol. 20, supplemental issue, 2005, pp. 25-46. Desmet, et al. discuss the initial implementation of the <emma> project, a markup-based writing workflow for students as well as an e-portfolio platform intended to collect student writing across their coursework and careers. Its successor platform, eLearning for Writers, continues to offer a markup function via more of a GUI-driven, highlight-and-click style interface, though overall it aspires more to the e-portfolio platform and learning management system with writing-pedagogy inflections (e.g., a function to compare drafts of an essay). According to the University of Georgia Department of English's 2021-22 First-Year Writing Guidebook, "eLearning for Writers (eLW)...is designed for composing, collaborating on, and revising academic writing. Created from FYW's previous platform, Emma, eLW offers a range of tools for writers and instructors. These include journal and brainstorming space, draft organization and comparison, and the ability to create comprehensive peer and instructor reviews at various stages of the writing process. Peer partners and instructors can provide holistic feedback, insert comments into a document, or use markup tags to identify common writing issues" ("First-Year Writing Program Guidebook").

[117] See Wendell Piez, "Beyond the 'Descriptive vs. Procedural' Distinction," *Markup Languages Theory and Practice*, vol. 3, no. 2, 2001, pp. 141–172.

below respectively show the student-generated genre model for a movie review and student-generated markup applying that model to a text in Smith's XML-based writing course.

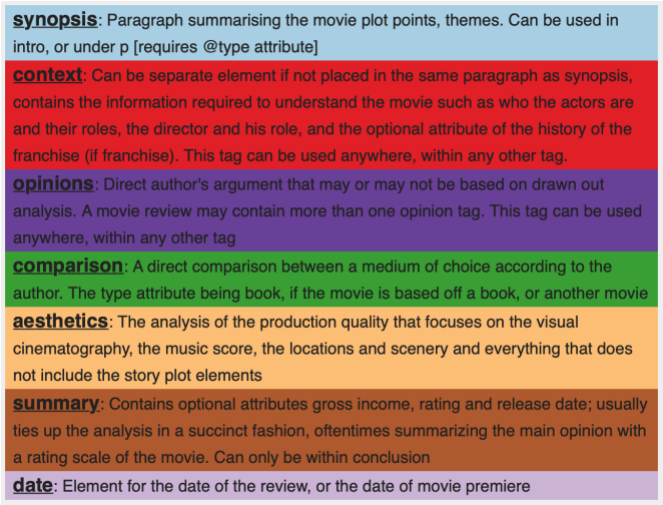


Figure 16: student-generated rhetorical genre model of a movie review from Kevin G. Smith's XML-based writing course, featured on his *Markup in the Writing Classroom* website (screenshot detail)

```
<p>
  <context type="director">Being only the second film directed by former dance
                choreographer Kenny Ortega, it is understandable that he would have wanted
                to go all out, so to speak, with the special effects in this movie in order
                to give it a "wow" factor.</context>
  <aesthetics>For all of the flashiness of witches flying on broomsticks, a CGI
                talking cat, a book with a living eye, and the process of turning into a
                stone gargoyle and exploding into a million pieces it feels like there is no
                substance to the effects; they never brought me into the film, and it seems
                they were generated solely for the sake of existing. </aesthetics>
  <opinions type="critique">In addition to this, there is little in the way of a
                coherent story — it consists mostly of people running around connected only
                by a series of thin plot devices. The movie changes tone as casually as the
                actors wear their costumes, and
                doesn't hesitate when breaking its own mood altogether (as when the three
                witches suddenly perform "I Put a Spell on You" directly after a somewhat
                intense chase scene). </opinions>
</p>
```

Figure 17: excerpt of student markup applying the student-generated schema for a movie review, featured on Kevin G. Smith's *Markup in the Writing Classroom* website (screenshot detail)

Smith's charge, in essence, concerns where the center of the classroom is located. Is it the instructor or the students? Is it the teaching or the learning? In addition, his critique implies an

assessment of learning outcomes as he positions application against creation, the latter sitting at the top of the traditional Bloom's taxonomic pyramid. Smith's contributions are significant in the small body of literature on markup in writing studies and composition, particularly for the focus they bring on the potential for markup language to attune students to writerly moves in relation to the (negotiable) boundaries of genre. Students often struggle with rhetorical frameworks for understanding and engaging genre in writing-based activities: what kinds of things should I say and do in X type of assignment? what makes a paper a certain kind of paper (research, review, etc.) as opposed to something else? where is the overlap between writerly agency and readerly expectations in terms of adopting and adapting genre for specific purposes? While Niester-Mika and Smith find their course designs to take better advantage of the pedagogical affordances of markup, I would place their efforts along with mine in a larger ecology of curricular goals and learning outcomes in the humanities that code, broadly conceived, allow us to view in new and striking ways. Smith's approach to markup—and, importantly, his approach to first-year writing in general—privileged the awareness and manipulation of genre as the primary goal for student learning. The call to teach with markup in more constructivist and collaborative ways, however, influenced the course design and pedagogy of the digital editing and text encoding course that I examine in the next chapter (taught five and six years after this chapter's first-year writing course). How might those students make design-level decisions and work together at all stages of a project-based course to model their work and iteratively apply and refine those models with the end goal of producing a digital documentary edition of historical correspondence?

Concerning the first-year writing class, additional limitations point to future possibilities for markup-based writing and composition. A more refined use of the TEI framework with ODD

(one document does it all) files would better position students to leverage the markup schemes in their writing and better position the instructor to design markup schemes for assignments.[118] The schemes suffered, for example, from an overreliance on the `<seg>` or arbitrary segment tag with a lengthening list of `@type` attribute values where other tags may have been more strategic. The `<seg type="evidence">` tag, for instance, was not as intuitive as `<evidence>` would have been in the context of a writing course in which students simultaneously composed *and* encoded texts with the primary goal of learning collaboratively rather than preserving, editing, and publishing a digital edition of extant text(s). This mismatch of goals meant that most of the TEI's almost-600 elements were irrelevant for the course. Overall, it is unclear if the TEI framework represents the best possible approach for XML-based writing courses given that the TEI lexicon is aimed at the interoperability of text encoding efforts whereby (usually) non-digital documents are transcribed and enhanced with metadata for publication as digital editions or collections.

The TEI framework nonetheless provided a robust model and authoritative language for understanding and describing texts at both the macro and micro levels, while the workflow of the writing course presents another area for further inquiry. Does the <oXygen/> XML editor, for example, present the best interface for students to compose their texts? Aside from more obvious issues of usability (e.g., the admittedly error-prone process of merging edits to the corpus file in the subversion repository), exactly how much does it matter that students *write code* as opposed to just thinking more explicitly in ways that the code allows? Developers have begun to produce software platforms aimed at preserving the intellectual processes that markup enables without the

[118] The TEI manages Roma, an ODD editor, at romabeta.tei-c.org/. According to its home page, "Roma enables you to create a customization of a larger scheme such as the TEI. It provides a user-friendly interface to pick and choose Elements, Attribute Classes, Model Classes, and Datatypes used in a schema. For each element the documentation, attributes, class memberships and content models are able to be modified" (retrieved 18 Dec. 2021).

need for using markup language per se in the way that one does in a code editor like <oXygen/>. The home page for Performant Software Solutions's FairCopy editor, for example, describes the software as "[a] word processor for the digital humanities" designed for non-experts who want to transcribe and edit documents "without writing a single line of XML." The website analogizes FairCopy with Microsoft Word and Wordpress, describing itself as "a 'semantic word processor,' a tool for capturing and organizing the layers of meaning encountered in textual artifacts." The interface combines aspects of word processing programs with drag-and-drop GUI functions to facilitate the manipulation of TEI-compliant XML. Figure 18 below shows the FairCopy interface with stacked purple boxes representing the hierarchy of encoded objects.

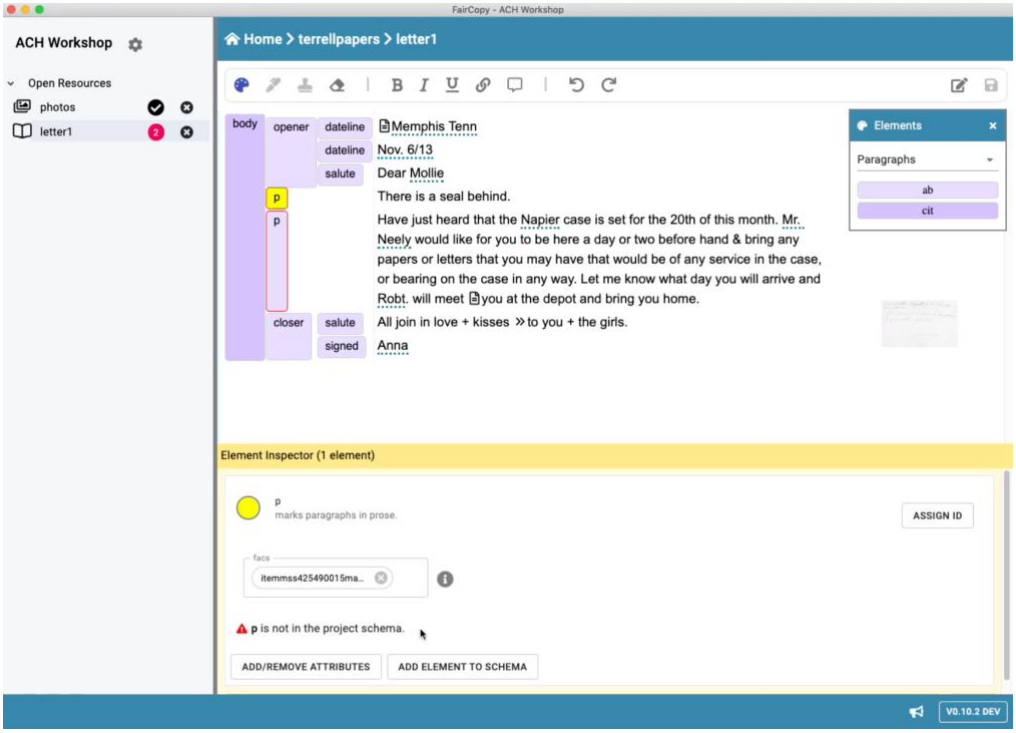


Figure 18: screenshot of FairCopy demonstration video "FairCopy v.0.11" (Laiacona)

FairCopy also allows for TEI customization through similar interface formats as well as side-by-side ‘encoding’ with digital facsimiles of documents. Figure 19 below shows this dual view with a click-and-drag interface for identifying and marking up zones in the digital facsimile.

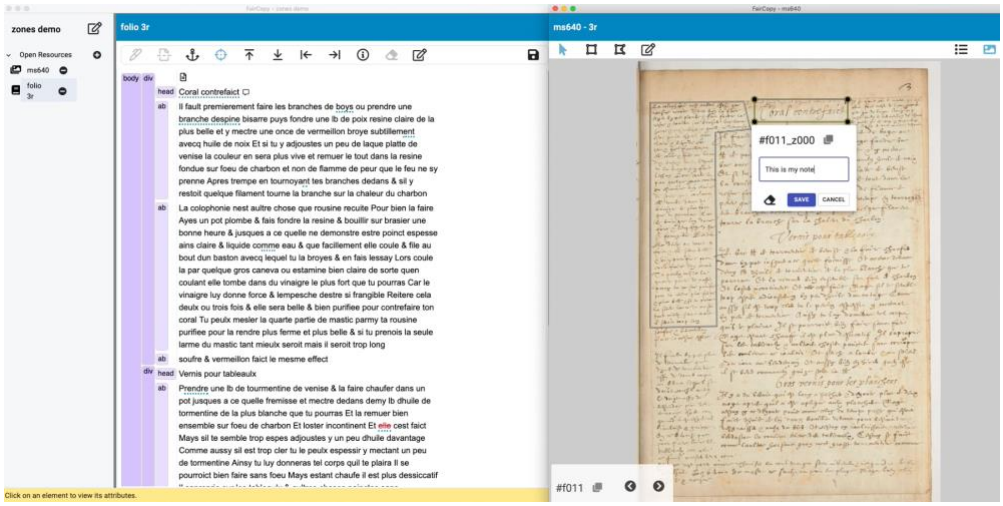


Figure 19: screenshot of FairCopy demonstration video “New Features in FairCopy v0.8.0” (Laiacona)

The latter function is more significant for the case study in the next chapter, but both examples of FairCopy’s interface point to the larger question of whether the kind of thinking prompted by writing (in) code—specifically, XML—can be replicated effectively without the need for ‘code’ per se. Considering this question, I return to Niester-Mika’s exhortation to involve students in more explicitly liberatory and interventionist learning experiences and note here that any layer of mediation conceals to greater or lesser degrees how digital platforms and software influence user thinking and behavior as well as how they perform actions that the user commands (cf. Stiegler’s notion of proletarianization—or, to give it even more of a Marxist inflection, the alienation of the user from the means of digital production). While FairCopy presents an otherwise innocuous example of digital mediation, I sought to cleave to the spirit of ‘seeing’ (and writing) the code itself in the course design for this first-year writing class.

Further directions for research would involve a mixing of social sciences and humanities methodologies to assess the effectiveness of various encoding frameworks including the TEI as well as more custom efforts. Kevin Smith describes such an approach as "teacher research that is supplemented with rhetorical theories of usability and participatory design" (4). With methods grounded in educational psychology and learning science combined with best practices in writing program assessment, a larger research program may identify the curricular and pedagogical elements—as if elements of an educational markup scheme—that exert the most impact while presenting minimal barriers for student learning. Potential applications range in scope from the individual course, as in this case study, to program-wide or even institution-wide deployments as in the University of Georgia's <emma> project (see note 116). Multi-institutional data, of course, would enhance the validity of any future findings. It would also be important to investigate the writing curricula and assignments best suited for a text-encoding approach; I've noted that this chapter's experimental case study did not seek to alter the existing curricular framework of the course, preserving the learning outcomes, assignments, topics, and schedule of the standard first-year writing course at The Ohio State University. Were a writing or composition course to use XML, what ideally would that course try to do, how would it be organized, and what would it ask of its students? As an experimental effort, the course described in this chapter suggests the potential for markup-based approaches in writing courses to enhance students' meta-awareness about composition and engage them in the slow, reflective thinking that seeing the code enables.

Chapter 4: Text Encoding, Open Pedagogy, and Public Digital Humanities

Editing undergirds moral and social order, and is in effect social work, work to facilitate knowledge exchange.

> *Martha Nell Smith, "The Human Touch, Software of the Highest Order: Revisiting Editing as Interpretation"*

While we should continue to explore tool building, visualization, and data mining as crucial areas within digital humanities, the narrow digital canon should remind us why we cannot stop digital edition work.

> *Amy Earhart, "Can Information Be Unfettered? Race and the New Digital Humanities Canon"*

<Introduction/>

While the case study in the previous chapter examines the experimental use of markup language to compose texts in a first-year writing class, the case study in this chapter looks to the more established practice of using extensible markup language to transcribe and edit extant texts to create digital archives, collections, and published editions of archival material.[119] The overall goal of the chapter is to extend the inquiry of the dissertation—*what is the value of code, and what are its uses and impacts for humanities education?*—into a different pedagogical context within English studies that, nevertheless, remains focused on the production and reception of texts, i.e., archival work, critical and documentary editing, and digital publishing. The notion of visibility behind seeing the code turns outward in this instance as instructor-student teams begin

[119] A modified form of this chapter is forthcoming in the *Palgrave Handbook on Digital and Public Humanities*, edited by Anne Schwan and Tara Thomson, Palgrave Macmillan, 2022.

within the physical confines of archives and special collections and complete their work on the open-access world wide web. As Matthew K. Gold describes it, the digital humanities 'moment' in general was marked by a sense of sudden visibility and public interest at mainstream scholarly venues (e.g., the MLA annual convention) as well as among emergent discourse communities on the web and social media. For Gold, this moment signaled an irrevocable turn toward a future in which the digital humanities as well as the scholarly productions of its practitioners would be objects of public attention and debate. In the opening chapter of Gold's inaugural *Debates in the Digital Humanities* collection, Matthew Kirschenbaum speculates that

> the digital humanities today is about a scholarship (and a pedagogy) that is publicly visible in ways to which we are generally unaccustomed, a scholarship and pedagogy that are bound up with infrastructure in ways that are deeper and more explicit than we are generally accustomed to, a scholarship and pedagogy that are collaborative and depend on networks of people and that live an active, 24-7 life online. ("What is Digital Humanities?")

Since Kirschenbaum's hopeful pronouncement, scholars have complicated the alleged arrival of the digital humanities in the academic mainstream, especially with respect to the availability of educational and career opportunities, the diversity and inclusivity among practitioners, and the politics and representation of knowledge production. Ted Underwood, for instance, writes that "[t]here is no entrance to this field. What we have is more like a door painted on the wall." His critique mainly concerns the absence of comprehensive (and available) educational opportunities for undergraduate and graduate students, as well as the ersatz reliance on one-off workshops or extra-disciplinary collaborators to shore up gaps in knowledge. "So of course the field tends to

attract people who already have an extracurricular background—which, of course, is not equally distributed. It shouldn't surprise us that access is a problem when this field only exists as a social network."[120] Beyond issues of training and education, Roopika Risam emphasizes how "axes of identity and the variable levels of privilege and oppression they confer shape scholarship, theory, method, and project design within digital humanities" (*New Digital Worlds* 24). She surveys a longstanding critique with entries such as Tara McPherson's "Why Are the Digital Humanities So White?" which investigates why scholarly considerations of "tools and infrastructure" tend to remain silent on issues such as "immigration, race, and neoliberalism." Among the exhortations that McPherson makes in her concluding remarks, she urges us to "engage the vernacular digital forms that make us nervous, *authoring* in them in order to better understand them and to recreate in technological spaces the possibility of doing the work that moves us." In the pedagogical spirit of this dissertation, I would add that teaching as well as authoring in digital forms—perhaps even teaching *via* authoring in them—opens up possibilities for meaningful humanities work with/in code. Risam also cites Amy Earhart's inquiry into race and the 'new' digital humanities canon, in which she frames the act of "building editions and digital texts as an activist intervention" into canons of knowledge that remain "closed" despite other scholars' utopian pronouncements about the democratizing effect of hypertext, the web, and digital publishing. While the leading edge of digital humanities scholarship seems to have moved on to inquiries about big data and analytics, Earhart emphasizes that "we cannot stop digital edition work" for its role in developing a better understanding of cultural heritage via "digital textual recovery." Technology historian Mar Hicks

[120] In particular, Underwood cites the dearth of curricular attention to social science research methodologies, which have become integral to contemporary digital humanities work that involves large-scale datasets, qualitative and quantitative analytics, and specialized research software.

would describe this work as a commitment to a "labor of care" (in her argument, for COBOL and other legacy coding infrastructures) in the face of the neoliberal fetish for disruptive innovation. To distill the provocations and critiques of Kirschenbaum, Underwood, Risam, McPherson, Earhart, and Hicks, it is critical for a field rife with proclamations concerning its transformative potential that we continue to ask what we are making visible, whom we are inviting into our collaborative and networked relationships, and how our work engages different publics and positionalities.

This chapter offers one response to these questions by considering what it means for a digital humanities *pedagogy* to be publicly oriented around/through code. Kirschenbaum initially "bracket[s]" teaching and learning as an aside to the scholarly inquiries and production of digital humanists (cf. the quote that opens this chapter), while I attempt to center pedagogy following calls (themselves emerging from the digital humanities moment) to consider teaching and scholarship with a sense of intellectual equity and, even further, to deconstruct the artificial binary between the two (Hirsch 5). As part of that intervention, I also attend to scholarly context in addition to recounting particular activities and the use of digital tools to avoid the lightweight approach to digital (humanities) pedagogy identified by Cheryl Ball, John Tinnell, and others ("Logging On"; "Post-Media"). As Amy Earhart writes, particularly concerning the production of digital scholarly editions, "we have much theoretical work to do in the selection of materials and application of digital tools to them." Applying her exhortation to teaching, I take as a focus one of the more ubiquitous cases for digital humanities work as public scholarship: digital editions that make otherwise difficult-to-access archival materials available for public viewing and study via open access on the web. Early and large-scale projects such as the *William Blake*

Archive, the *Rossetti Archive*, the *Walt Whitman Archive*, and the *Women Writers Project* often appear as milestones of public scholarship in historiographies of the digital humanities, but their scale presents a considerable barrier for pedagogical approaches that would involve students, especially undergraduates, as decision-making agents throughout the larger intellectual journey that such a project represents: from archive to encoding and editing, to design and publication.

As a way of engaging students in text encoding and public scholarship I turn to the local context of institutional archives and small-scale, curated, "boutique" projects that attend to the methods and meanings that go into producing, processing, and presenting qualitative sources and datasets that would be otherwise unknown and/or inaccessible as collections (Ball, Graban, and Sidler 5; Crawford 9–11). These projects draw from Johanna Drucker's distinction between data and capta, an epistemological shift that emphasizes the constructed nature of knowledge as opposed to encountering datasets as *a priori* representations for computational analysis. In other words, and especially in a pedagogical context, we should avoid "collapsing the critical distance between the phenomenal world and its interpretation" by engaging data as context-dependent acts of interpretation. We ask students to see the data as they would see the code, and, even further, to see text encoding as the creation of data that opens up space for more interpretations and so on. Matthew Lavin has recently offered an emendation to Drucker's intervention by etymologically and conceptually rehabilitating data via "feminist and/or humanistic" frameworks established by Catherine D'Ignazio and Lauren F. Klein in *Data Feminism* as well as Katherine Bode in *A World of Fiction*.[121] Lavin advocates that we understand data as "situated knowledge"

[121] Catherine D'Ignazio and Lauren F. Klein, *Data Feminism*, MIT Press, 2020; Katherine Bode, *A World of Fiction: Digital Collections and the Future of Literary History*, University of Michigan Press, 2018.

that seeks to measure, describe, or explain phenomena in ways that are inextricable from their social constructions and contexts. Using "tamed data," Andrew Goldstone argues, allows students to engage data's complexities and ambiguities along "a trajectory from exploration to valid argument," while unwieldy "data in the wild" often presents barriers for courses that intend for students to have coherent and intellectually rewarding experiences. While Goldstone focuses on the teaching of quantitative research methods, I apply his insight to qualitative projects that involve textual and historical data and artifacts. In pursuit of that goal, this chapter first reviews pedagogical engagements with the archive, which tend to fall into three categories: practice with research methodologies, encounters with primary sources, and activities with curatorial goals. Relatively underdeveloped, however, are frameworks for student partnerships and project-based learning with the institutional archive. Drawing from precedents in rhetoric and composition, library collaborations, and digital humanities, I propose an open pedagogical approach for the creation of public-facing, publicly engaged digital resources. As a model for operationalizing this approach I review a course that scaffolds student learning from the curation of archival materials to the publication of an open-access, digital documentary edition of historical correspondence.

I taught this course during the 2018 and 2019 spring semesters as a section of Honors 301: Proseminar (subtitle: "critical editing and publishing in the digital humanities") in the Lewis Honors College at the University of Kentucky. According to the Honors College course catalog, HON 301 is

> [a]n interdisciplinary seminar that offers an in-depth examination into a topic chosen by the instructor, related to history, culture, and constructions of knowledge. The format of the seminar is discussion-based and student-centered. Throughout the semester, students

> will be responsible for leading assigned discussion in collaboration with their peers and/or professor. This course enhances critical and creative thinking skills through specialized projects, research, written assignments, and/or oral presentations. ("Honors Courses")

On the one hand, the course represents what Robert K. Yin calls the "common" case study for its similarity to extant pedagogical efforts in text encoding and digital scholarly editing. On the other hand, the course also offers an unusual case given that it was offered at the undergraduate level rather than the graduate level and sought to involve students in all aspects of the editing process from making decisions in the archive to making decisions on the webpage. The case in particular shows how the affordances of code as a revelatory medium prompt us to match its rhetorical and hermeneutical capabilities with a curricular and pedagogical design that centers student learning.

<Students and the Archive/>

"I believe pedagogical work should be honored as the best kind of research, and our scholarship should be pedagogical," Jesse Stommel writes in *Disrupting the Digital Humanities.* For Stommel, the disruptive (and public) potential of digital humanities lies in "mak[ing] the work legible" so that students, teaching-focused faculty and institutions, and broader audiences are invited into the process of knowledge production in "a more intimate, more provocative" way (84). Engaging students in open pedagogy and the archives certainly makes the work more legible and accessible for them, but it also invites them to make the work legible and accessible

for others. In other words, students stand to do more than *use* archives; their greatest potential lies in partnering with archivists, instructors, and other stakeholders to participate in the ongoing project of refining, assessing, revising, and leveraging our historical imagination. While students themselves represent one 'public' in this network, they also act as a conduit through which the archive and the academy at large can address audiences in other professional organizations and contexts as well as among the lay community.

There remains a significant opportunity to develop pedagogical frameworks and research on institutional archives and public scholarship in the digital humanities. Teacher-scholars have addressed the archive in a variety of ways, including how archival spaces and materials offer students an opportunity to practice research methodologies central to humanities disciplines. One example from rhetoric and writing—notably, a graduate-level course—offers a representative dyad of learning outcomes: "to expose students to the practical aspects of working in an archives facility" and "to provide them with opportunities to practice arguing with and about archived material" (Buehl, Chute, and Fields 281). Others have explored the use of digital archives to foreground information literacy and primary sources in undergraduate courses where students might otherwise encounter only secondary material in textbooks, modern editions, and other mediated/mediating resources. Jessica Enoch and Pamela VanHaitsma advocate for a focus on "archival literacy" in courses that ask students to make observations and arguments from digital or digitized materials. Emphasizing the "rhetorical characteristics of digital archives" leads students to consider "the archive's power, its promise, and, indeed, its problems," they write, lending more nuance and a critical mindset to student work with archival sources (218-19). In other words, we ask students to consider archives not only as repositories of information and

artifacts for their learning, but as constructed spaces that involve human decisions and exert real agency in the world towards particular goals or ends.

The nature of archives, of course, remains contested, and scholars have examined the "institutionalization" of the archive in terms of the actors, purposes, and interests involved in creating, maintaining, and providing access to archival collections housed in educational, museological, cultural heritage, and other institutions (Manoff 19). Archives, then, represent not only material for students to leverage in pursuit of their learning, but perhaps more importantly they represent sites of critical inquiry following a contextual and *rhetorical* understanding of data (cf. Drucker, Lavin, and the discourse on data, capta, and situated knowledge) as both product and agent of persuasion. Digital humanities projects may describe the curation and exhibition of materials as the construction of an archive, but Kate Theimer reminds us that a less fungible and more infrastructural sense of archival work implicates the acquisition, metadata, organization, access, storage, and sustainability of entire collections as opposed to ad hoc curatorial agency. This latter pursuit, however, represents a third branch of archival pedagogy that engages students in the assembly of collections or exhibits that convey inquiries, messages, or arguments relevant to a course's learning goals. Using content management systems such as Wordpress, Omeka, or Scalar, students may construct open-access websites where they articulate their insights and organize what they have selected from archived or found materials (Tanaka et al. 42). Students may also collect or generate records in the field by way of "pop-up" endeavors with the goal of "enacting the work of archiving," which involves students in the networked relations and actions between archivists and subjects *in situ* (Rice and Rice 251). These efforts may be freestanding in a single course or attached to the development of larger projects; for

instance, assignments linked to the Digital Archive of Literacy Narratives have asked students to locate subjects and record interviews in their local communities, which are uploaded to the archive (Comer and Harker 71).[122]

To develop a pedagogical grounding for digital humanities in the institutional archive, particularly institutions of higher education, I draw from Wendy Hayden's (2017) notion of "the archival turn's pedagogical turn" in courses that engage students in "inquiry-based model[s] of education" across the disciplines (135). The specific activities and goals of these courses differ, but Hayden finds common ground in an emphasis on "collaboration, invitation, locally-based research, and activism" with "outcomes of recovery, rereading the archives as a source of knowledge and public memory, and archival creation by students." Owing to the central role that uncertainty plays in archival research, a pedagogy of the archives leads students to see their learning less as "knowledge accumulation" and more as the "inquiry and knowledge production" of scholar- and author-apprentices (145). In this sense archives function less as textbooks and more as laboratories that invite students—and instructors—to experiment in and with a publicly visible, evolving, and contested space. We see these principles at work in collaborations that treat the archive as "a site to experiment with new ideas, methodologies, projects, and pedagogies" such as the University of Southern Mississippi's Save Our Stories project, an open-ended call for essays based on items in the archival collections (Branock, Carey, and Inman 168). Flexible to the needs of individual courses and open to all genres from the creative to the critical, Save Our Stories enacted a partnership with distributed authorship and agency for the stories told

[122] Founded in 2007, the Digital Archive of Literacy Narratives is "an open public resource made up of stories from people…about their experiences learning to read, write, and generally communicate with the world around them." DALN materials "can be text, video, audio, or a combination of formats." See https://www.thedaln.org/#/home.

by and about the materials in an academic, institutional archive. As told by USM's librarians, the aim was to "expand the umbrella of digital humanities to include...the countless forms of pedagogical and bibliographic work that go into the curation and contextualization of archival materials for the broader public" (164). In broad terms, Save Our Stories sought to prompt interest in archival work and collections, to cultivate students' historical imaginations, to enrich the widest possible range of courses at the institution, and "to reframe the digital humanities from the perspective of faculty, librarians, and students" (165).

Other pedagogical collaborations with the archive pursue more specific projects at the course level, resulting in more sustained engagement with a particular collection or materials as well as with particular scholarly issues and methods in the digital humanities. One of the more storied activities in digital humanities scholarship is the digitization, enrichment, and publication of digital editions of historical or literary texts for open access on the web. While the production of digital *archives* represents a more sprawling, long-term (or ceaseless) endeavor often centered on an author's body of work—for example, *The Charles W. Chestnutt Archive* or *Digital Mitford: The Mary Russell Mitford Archive*—digital editions offer more granular projects with a focus on collections with more defined or limited parameters as well as specific texts or documents.[123] The Romantic Circles Electronic Editions site offers a range of examples from *Anna Letitia Barbauld: Letters to Lydia Rickards, 1798-1815*, a collection of 38 letters, to *Mary, the Osier-Peeler*, a digital edition of a single, 320-line poem.[124] When it comes to involving

[123] *The Charles W. Chestnutt Archive*, edited by Stephanie P. Browner, Matt Cohen, and Kenneth M. Price, chesnuttarchive.org/; *Digital Mitford: The Mary Russell Mitford Archive*, edited by Elisa Beshero-Bondar, digitalmitford.org/.

[124] Anna Letitia Barbauld, *Anna Letitia Barbauld Letters to Lydia Rickards, 1798-1815*, edited by William McCarthy, *Romantic Circles*, 2021, romantic-circles.org/editions/barbauldletters; Mary Morgan, *Mary, the Osier-Peeler*, edited by Emily D. Spunaugle, *Romantic Circles*, 2021, romantic-circles.org/editions/maryosier.

students in curating, encoding, and producing digital editions of archival materials, Clayton McCarl's "Editing the Eartha M.M. White Collection" at the University of North Florida is particularly instructive. Designed "as a partial internship," the summer course engaged students in a typical classroom format as well as "workshop sessions" for individual or small group work with the end goal of launching "a prototype website" along with a campus presentation of students' experiences (528-29). Students found the work energizing but also felt challenged by its "unpredictability" and "constantly evolving nature," which McCarl attributes to "the largely unstructured approach" of the course (533). The summer term's short time frame is not unusual for instruction in digital editing and encoding, which often takes place in workshops or modules ranging from a day to several weeks. Similar efforts have found that students need time and structure to reflect on their role as editors, to connect their work with an impact on stakeholders, and to understand how individual activities support larger course objectives (Duke and Stanley 65-68). With a more sustained academic term in mind, the following sections establish pedagogical orientations and lay out a scaffolding that attempts to balance structured learning and instructional guidance with student agency and the organic nature of research. Through the lens of public scholarship and the affordances of text encoding, students encounter archives as "a crucible of activity" where they are invited to negotiate and create knowledge in contrast to the passive materiality of sealed-off, "static" collections (Decker 238). In other words, we add to touchstone archival scholar "Randall Jimerson's analogies of temple, prison, and restaurant for understanding the purposes and powers of the archive with a fourth analogy: the workshop, to which the 'power of interpretation' extends to the assemblage, transformation, and (re) mediation of archival materials for a public audience" (Conatser, et al. 29).

<Pedagogical Orientations/>

Given the complexity of a course project involving archival work, text encoding, editing, historical research, and digital publishing for undergraduate students who are likely unfamiliar with digital humanities as a scholarly area in general as well as with its conceptual and technical methods, there is a temptation to devote a good amount of time at the outset of a course to cover content knowledge, i.e., the information, theories, and practices with which students ostensibly must be familiar to engage in the higher-order thinking that the project demands. However, this approach delays the project-based and collaborative learning that should characterize the entirety of students' experiences in the course, which ideally works more like a research lab or team than a class per se. To spur community and engagement around basic premises of digital humanities and text (en)coding, students may reflect on an approachable piece such as Kirschenbaum's "Hello Worlds" during the first class meeting. But the larger pedagogical approach involves what researchers have described as "the flipped, flipped classroom" (Schneider, Blikstein, and Pea) and the "flipped flip" or "co-creational" model (Uskoković). While a traditional understanding of the flipped approach, which has not been as novel for humanities pedagogy as it has for STEM disciplines, begins a lesson with lectures, readings, and other materials that students study prior to application-based exercises during class meetings, a further 'flip' of the format *first* engages students in activities for which they have not yet been systematically prepared. Other researchers have described this effect as "productive failure" insofar as students "consolidat[e]" the uneven, exploratory experiences of initial encounters with later instruction and learning activities as the class hones in on more optimal solutions, strategies, decisions, and deliverables (Kapur 289–90).

Part of the rationale for a co-creational design is merely practical; a single academic term does not provide enough time for students to gain a repertoire of knowledge and skills *prior* to conducting the project itself from conception to publication on the web. Even given adequate time to accomplish all of these tasks, however (e.g., during a year-long capstone), designing for productive failure stands to foster curiosity and motivation as students exercise agency in the course by considering unknowns and challenges, areas that merit further study or deliberation, connections between new and previous experiences, and adjustments of plans and schedule to accommodate new issues and insights. As Karen Cangialosi has argued for re-envisioning the pedagogy of basic STEM courses, "[o]ur primary role as teacher can be simply to create the best culture chambers for students to flourish" in learning environments that center inquiry and agency for projects that exceed students' content knowledge and familiarity with the discipline. Similarly, for undergraduate digital humanities courses, Ryan Cordell found that his impulse to introduce students to the field of digital humanities through thought pieces and polemics in the "what is DH?" genre failed as an attempt to establish a baseline familiarity with disciplinary concepts, methods, and issues because it did not draw from student interests, experiences, or prior knowledge. Put simply, the debates struck his students as the distant and inconsequential infighting of specialists with little connection to their work in the class. Broadly, his experiences suggest that attempts to scaffold introductory material and foster basic literacies may not escape the insider knowledge and affective orientations of the expert practitioner who is designing the curriculum. He recommends instead that we "start small" with experiential activities that scaffold towards larger objectives in ways that attend to the range of students' digital literacies and backgrounds. Experimental studies in psychology education support Cordell's sense that

"many texts and lectures presuppose a level of differentiated knowledge that is not available to novices," and that guided discovery via "contrasting cases" provides a basis for "a deep[er] level" of engagement and understanding with later explanations and activities (Schwartz and Bransford 504).

At a macro-level, the pedagogy of archival work, text encoding, and digital scholarly editing follows in the spirit of authentic assessment, an approach to assignment design and evaluation that eschews the testing or exercise of knowledge without context (e.g., on exams or in more traditional research-based writing assignments) in favor of involving students in tasks that are critical or common in particular fields of work (Wiggins). Authentic assessment involves realism, cognitive challenge, and frequent and formative feedback; however, the literature does not focus as substantially on decision-making processes, teamwork, and collaborative learning: a gap that project-based courses are poised to address (Villarroela et al. 844-46). One of the closest graduate-level cognates to the undergraduate course I describe in this chapter, for example, asked students to work individually on different projects (Engel and Thain). Beyond the pedagogical benefits of authenticity, ethical considerations frame the need to involve students in what Anne B. McGrail calls "the whole game" of digital humanities as opposed to merely "teaching them *about*" the field or treating them as "crowdsourced labor" that completes "microtasks" for projects in which they otherwise have no agency (emphasis original). Addressing the place of community colleges and their students in the digital humanities, McGrail warns that we will undermine the democratic potential of digital and public humanities pedagogy if we "reinscrib[e] students into their place in a hierarchy that is all too familiar" in the academy. My commitment to students, for example, is that they will work and be credited as members of the editorial team;

of course, this is easier said than done given the widespread habituation to gatekeeping culture in higher education as well as the delicate balance of instructional and student agency that enables student-centered, collaborative, and constructivist learning environments.

Lastly, though certainly not least, the design and teaching behind the chapter's case study takes a cue from open pedagogy, which leading proponents Robin DeRosa and Rajiv Jhangiani summarize as "an access-oriented commitment to learner-driven education" and "a process of designing architectures and using tools for learning that enable students to shape the public knowledge commons of which they are a part." Merging constructivist and critical traditions with the open access and open education movements, open pedagogy notably sees students *build* educational resources in addition to using them for learning. In an open pedagogical environment students come to understand that they are doing the work of researchers who publish digital editions, archives, and other resources; in terms of authenticity, then, the realism of the project is apparent. And, through the curricular scaffolding I describe in the following sections, students encounter cognitive challenges in ways that make space for exploration and productive failure, all the while building their capacity for evaluative (self-)judgment through the collaborative nature of the coursework. Overall I suggest that undergraduate curricula in digital and public humanities (via the curation, modeling, encoding, editing, and publication of archival materials) should engage students in a "non-disposable" project whose three dimensions are time, space, and gravity: the persistence of student work beyond the timeframe of an activity or course, the reach for external audiences or occasions beyond an instructor and class, and the "value/impact" for those audiences and students themselves (Seraphin et al. 89–90).

<Curation and Digitization/>

As a case study for how this pedagogical approach can be implemented as a scaffolded in a markup-rich curriculum I turn to "Digital Editing and Publishing," a junior seminar I designed and taught in the Lewis Honors College at the University of Kentucky with critical support and partnership from UK Libraries' Special Collections Research Center. Prior to the first offering of the course, I consulted with the director of research services and education on collections that aligned with my course goals and would present a compelling connection for local contexts and audiences. Approaching the special collections faculty and staff without a firm decision beforehand was important for the open (and public) design of the course. As a best practice for course design, "[working] with collections staff early in the course development process" helps to identify the best choices for materials and also provides an opportunity to develop "scaffolded approaches to working with primary sources" (Tanaka et al. 51). In consultation with Special Collections, I selected the Frontier Nursing Service collection: specifically, the correspondence of its founder, Mary Breckinridge, during 1919 when she worked in rural France as a nurse and administrator of relief efforts for women and children reeling from the devastation of the first world war. Scion of a wealthy and politically connected family from Kentucky, Breckinridge served on the American Committee for Devastated France and found inspiration in European models for providing nurse-midwife care to rural areas. Later bringing the first nurse-midwifery services to the United States, Breckinridge's life and work are significant for the history of nursing, public health, and Appalachian studies. Given our location at the doorstep of the Appalachian region where Breckinridge's nurses on horseback brought specialized medical care

to isolated communities, her work resonates as part of our local history and evokes the affordances and flexibility of place-based education in the archives (Beam and Schwier 13-16). In fact, the UK Special Collections' Breckinridge Research Room, where many of our class meetings were held, is named after her family. Because the Frontier Nursing Service Collection is large and contains a wide variety of documents dating from 1789 to 1985, I selected several folders' worth of material—letters written during Breckinridge's first year in France—for students to explore once the course began: i.e., enough to provide for meaningful student choice, but not so much that students were stymied by the sheer volume and breadth of material.

Following a co-creational design, it was important to get students in the archive from the beginning; after three meetings in the assigned classroom space we spent a month in the Special Collections Research Center. In addition to drawing from the flipped-flip model, I looked to Silvia Vong's sense of constructivist learning in the archive, which emphasizes "[c]onnection between prior knowledge and new knowledge; [s]ocial interaction between students or students and the instructor; [r]eal-life tasks or experiences that relate to new knowledge; and [a]ctivities that prompt reflection on one's own learning and experience(s)" (155-56). As Paul Fyfe suggests, this "unplugg[ed]" work in the archive prepares students to "appreciate, by contrast, their active mediation of similar work in the digital field, which too frequently seems transparent, or so flattened that students fail to notice its own critical topologies." In other words, "the insistent materiality" of archival objects leads students to understand how their later work manipulates and *transforms* texts in digital editions (Engel and Thain). Our time in Special Collections was devoted to studying the letters: piecing together the narratives they conveyed, refining our sense of their significance (biographical, historical, literary), and curating a selection that students

would transcribe, encode, and research for a digital documentary edition. Readings on archival theories and methods enriched the experiential learning while students also compared the letters with autobiographical and scholarly accounts of Breckinridge's life.[125] Class meetings began and ended with group discussions of progress and goals while much of the time involved a balance of individual work and team deliberation. In the spirit of constructivism, student observations and insights drove our work: what did they notice? what was compelling or confusing? what were others' reactions to the same document(s)? Students concluded the month in Special Collections by using specialized scanners to create digital facsimiles of their selected letters—one or two per student depending on length and complexity. Before they began to transcribe and encode in earnest, however, we turned our attention to how researchers model texts in digital environments to build a theoretical foundation for the detail-intensive work that would soon follow.

<Modeling Text and Correspondence/>

Arguing for the value of teaching computer programming in/as humanities coursework, Matthew Kirschenbaum eschews learning outcomes focused on coding skills per se—students can go elsewhere in the university curriculum for that, he asserts—as he recounts an activity that asks students to model snowballs in an object-oriented programming environment:

> What are a snowball's salient characteristics? What do you do with one? Well, you toss the snowball at someone else. But wait, before you do that, you first have to shape it, form it, pack the snow. Once you do toss it, do you still have it? No. So the program has

[125] See the syllabus in Appendix B for the list of readings embedded in the class schedule.

> to be able to distinguish between possession and nonpossession of the snowball. And maybe, if you hang onto it too long, it starts to melt. The exercise of thinking through what it takes to model a snowball in a believable fashion goes a long way toward capturing the appeal of what I mean by programming as world-making. ("Hello Worlds")

Modeling is a core activity of digital humanities scholarship often defined as an interpretive and transformative pursuit (e.g., as world-making). In the foundational monograph *Humanities Computing* Willard McCarty describes modeling capaciously as "a representation of something for purposes of study, or a design for realizing something new" (24). At the same time, the novel possibilities of modeling are always in tension with "the fundamental dependence of any computing system on an explicit, delimited conception of the world or 'model' of it" (21). Computer code thrives on "univocal, correct, 'activating' interpretations" and cannot work with the "ambiguity, context, and polysemy" that are a fundamental condition of human thought and language (Golumbia, *Cultural Logic* 84). In other words, nothing goes without saying when modeling real-world phenomena for and with computers, but for humans, models will always leave something out and will not serve equally across every context or application. However, it is this very constraint that digital humanists find to be generative, especially for pedagogy; in deciding what to include, how to include it, and what to leave out when modeling concepts, actions, and other phenomena, students engage in a "mental discipline" that "self-reflexive[ly]" discovers, questions, and "make[s] explicit our understanding of the thing being modeled" along with "our assumptions and beliefs about it" (Sperberg-McQueen 2019, 287-88). Per statistician George Box's dictum, "all models are wrong, but some are useful" (Box and Draper 422).

My goal for modeling exercises is that students practice a kind of thinking unique to text encoding, where instead of snowballs, scholars model documents, language, writing, and social and historical context. Our work focused on correspondence, which presents a unique test case given its complex nature as a document and mode of communication. Text encoding and digital editions are particularly adept at addressing the challenges of correspondence given that (1) letters are often held in multiple, "far flung" locations with an "equally dispersed" audience of archivists, scholars, commercial interests, and lay enthusiasts; (2) print editions cannot contain contextual and associated artifacts that defined the social networks in which letters operated; and (3) the study of letters in codex form "essentially reduces and thus misrepresents the relational aspect of correspondence" (Hankins). In short, correspondence presents the challenge of modeling texts that gesture outward or elsewhere, and students explore how the computational and representational capabilities of markup technologies address the nonlinearity, referentiality, and contextuality of correspondence. This work engages students with a kind of writing that they likely use regularly however modernized their own media and conventions are by comparison.

As an initial exercise I asked students to model correspondence using only a wall in the classroom and post-it notes. The task was deceptively simple: create a model of correspondence that is as comprehensive and accurate as possible. As a heuristic, I provided a list of generative questions: who is involved in correspondence? how is it structured? what does it contain? how does it move across space and time? what are its physical aspects? what documents or objects are associated with it? what do letters tell us, and what would we want to do with them? Initially, students generated as many ideas as possible on their own; each went on a post-it note. Next, they worked together to cluster their notes around common themes on the wall. Once an initial

affinity map was complete, students reflected on what was represented most and least, which categories or affinity areas needed to be combined or better distinguished, and what still was absent from the emerging model. After this discussion the class attempted to categorize and build relational hierarchies with the goal of constructing a tree diagram of the correspondence model. Naturally, students disagreed, leading to further conversation, while particular items seemed to fit in several places or in no place at all. These moments served as useful reminders that the exercise seeks to engage students reflectively in the process and choices that go into modeling a text and document for a variety of purposes. For example, a researcher seeking to understand letters as a social network would conceptualize them differently than a researcher approaching them as historical records.

For the next meeting students studied an article recounting the Text Encoding Initiative's development of a model for encoding correspondence as part of the TEI Guidelines. Recalling their experiences in the same task, students considered the theories of correspondence that the authors review and model in extensible markup language. Letters are at once object, text, and event, the authors conclude, each of which implicates people, dates, places, and sequences. The TEI task force ultimately arrived at a "communication-oriented concept of correspondence" with tags for action and context in an overall description wrapper (Stadler, Illetschko, and Seifert).

```
<correspDesc>
  <correspAction type="sent">
    <persName>Carl Maria von Weber</persName>
    <settlement>Dresden</settlement>
    <date when="1817-06-23">23 June 1817</date>
  </correspAction>
  <correspAction type="received">
    <persName>Caroline Brandt</persName>
    <settlement>Prague</settlement>
  </correspAction>
```

```
  <correspContext>
    <ref type="prev" target="http://www.weber-gesamtausgabe.de/A041209">Previous letter of
      <persName>Carl Maria von Weber</persName> to <persName>Caroline Brandt</persName>: <date
      from="1817-06-19" to="1817-06-20">June 19/20, 1817</date>
    </ref>
    <ref type="next" target="http://www.weber-gesamtausgabe.de/A041217">Next letter of
      <persName>Carl Maria von Weber</persName> to <persName>Caroline Brandt</persName>: <date
      when="1817-06-27">June 27, 1817</date>
    </ref>
  </correspContext>
</correspDesc>
```

With guiding questions, students arrived ready to compare the TEI model with their own, as well as with the precedents that the TEI task force considered. We discussed how and to what extent the `<correspDesc>` model is, as its developers write, "both theoretically justifiable and practically useful" in the overall context of the TEI Guidelines (Stadler, Illetschko, and Seifert). Why, for example, would it be desirable to minimize the creation of new elements to action- and context-oriented tags aside from the overall wrapper? How might modeling textual data in this way allow for interchange and interoperability with other researchers and projects? Beyond those, why do models and metadata like this matter for a public audience? The TEI's model should not present a foregone conclusion at which students must sooner or later arrive; instead, it offers students an opportunity to consider the needs of their own project and, in the spirit of open pedagogy, decide if customization is needed. Students may also consider alternative models such as Dumont et al.'s proposal for the Correspondence Metadata Interchange Format (version 2), which includes a unique witness identifier (i.e., "references to the underlying archival document as well as to other editions of the same letter"), editorial certainty (e.g. concerning authorship, recipient, scribe, and so on), entities mentioned (e.g., people, places, events, objects), and type of publication (e.g., "from simple archival repertories to regesta to fully edited letters which include commentaries and facsimiles").

The activity concluded with students encoding their letters' metadata as `<correspDesc>` structures in the project corpus file either in person or as homework, connecting theoretical modeling to encoding practice as a scaffolded, experiential learning activity. Overall, as a result of this sequence, students were able to describe the importance of modeling for text encoding and digital scholarly editions, as well as evaluate the affordances and constraints of various models and modeling approaches. These skills were critical for a mid-term assignment (during the first iteration of the class) that asked students to select, analyze, and evaluate a published digital scholarly edition of correspondence and present their insights to the class (see Appendix C). Through this assignment, students applied the same modeling methods to digital scholarly editing and electronic editions and used those insights to make decisions regarding the eventual 'published' form of Breckinridge's correspondence.

<Transcriptional and Contextual Encoding/>

For approximately two months students transcribed, encoded, and edited their selected letters. As is standard for digital-textual editing projects in the humanities, we used the Text Encoding Initiative Guidelines for XML encoding. For a full discussion of the history, uses, and technical aspects of both XML and the TEI Guidelines, see the previous chapter (particularly, the sections "Markup Language and XML" and "XSLT and the Text Encoding Initiative"). Since the late 1980s the Text Encoding Initiative has refined a framework for modeling documents and texts in markup language—first using standard generalized markup language (SGML) and later XML—with a controlled lexicon based on document type. Prior to the Guidelines, text encoding

schemes “typically reflected the specialized interests of their originators and were, by and large, incompatible...a text encoded for one purpose or piece of software often required substantial editing to be used for another purpose or with other software, if it was reusable at all” (Ide and Sperberg-McQueen 5). Reading about and discussing the histories of markup language and the Text Encoding Initiative allowed students to understand (and, more importantly, to buy into) the value and purpose of the TEI Guidelines in terms of the ethos of public scholarship whereas requirements in other courses such as paper and citation formatting may strike them as arbitrary and inflexible aspects of educational assessment. As they gained experience and consulted the taxonomies and examples in the TEI Guidelines, students began to appreciate the “modularity, modifiability, [and] numerous alternative means of handling analogous phenomena” (Huitfeldt 176). In other words, the narrative of teaching and learning with text encoding involves the re-centering of student agency among what at first glance may appear to leave little room for self-determination (i.e., a prescribed list of terms). As my students and I reflect, “[w]e dedicated much of our time to identifying and applying the elements and attributes that would best serve our project, especially given the theoretical possibility that almost everything in a letter can be encoded in some way” (Conatser et al. 30). As in the correspondence modeling exercise, students placed TEI standards in conversation with the heuristic products of thought experiments as well as their increasingly sophisticated editorial and rhetorical understandings of the digital edition towards which our labor strove.

To scaffold student work, text encoding was split into two phases: transcriptional markup and contextual markup. Challenges may arise at the transcriptional stage based on the physical state of the documents as well as their compositional method. Our letters were relatively modern,

mostly undamaged, and with few exceptions written with a typewriter; while we did not need to engage in paleography or extensive handwriting analysis, the documents nonetheless presented many challenges for transcription: overtyped characters or words (sometimes twice or even three times over), faint or missing characters, handwritten emendations, and small tears. Collaboration was key, and we spent a significant amount of time as a community determining the best reading or interpretation for ambiguous moments in the text. Because our transcriptional markup sought to capture the diplomatic state of the correspondence—i.e., an accurate physical representation of the text/document as opposed to a 'clean' version that ignores what editors interpret as mistakes, deletions, revisions, superfluous marks, or unclear/unreadable passages—it was important for us to attend as a group to these ambiguous textual moments and reflect on our own interventions as editors when deciding on a best course of action for the transcriptional markup. Figure 20 below features detail screenshots of typical transcriptional challenges such as multiple overtypes and unclear handwriting.

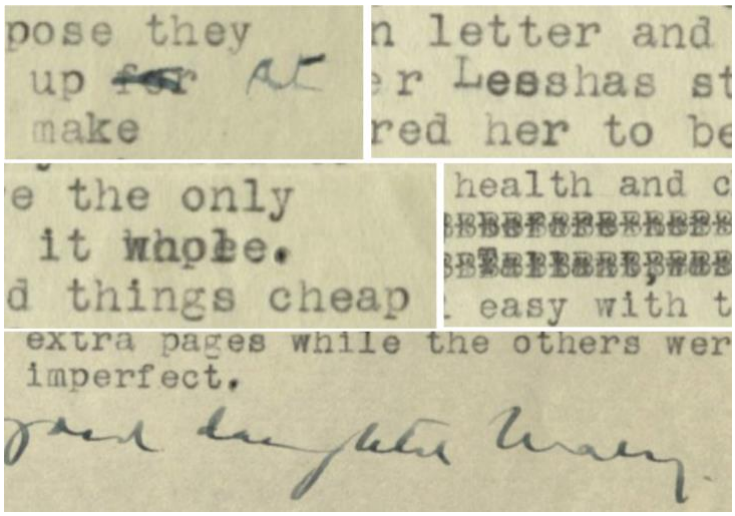


Figure 20: details from Mary Breckinridge's correspondence featuring transcriptional challenges (courtesy of University of Kentucky Libraries Special Collections Research Center)

Once the transcriptional markup was mostly complete, students began contextual markup, which involved tagging, researching, and describing named or referenced entities such as people, places, events, and organizations. Unique identifiers linked the contextual markup in the body text to fully descriptive entries in the metadata header for the corpus file. At this stage the markup itself took little time compared to the research and writing to flesh out Breckinridge's references, some of which veered into the obscure and led to spirited detective work on the students' part. Breckinridge was well connected in both Europe and the United States with a significant social network and frequent travel habits. As with all aspects of markup, we discussed the best approach to modeling these contextual elements given the purpose of our edition as well as the purposes our readers may bring to it. The work and thinking required to flesh out these elements was unique for students who were accustomed to dealing with only secondary literature as 'research.' More familiar and modern secondary sources such as twenty-first century scholarly monographs were placed alongside contemporary and/or local accounts of culture, events, and other phenomena. While the course did not primarily focus on history-based methodologies, students nevertheless practiced a historian's work and mental moves while piecing together evidence and scholarly analysis to better understand the significance of the documents before them. Seemingly incidental or personal references took on a larger resonance as, for example, Breckinridge's visit with a "radical" baron led us to consider the larger political context in which Breckinridge conducted her relief work, e.g., the nationalism of the right-wing Bloc National in France after the war.[126] These contextual tags in the transcriptions were linked to header entries

[126] This particular example, too, speaks to the historical 'weed-work' students engaged in given that the baron's wife, Louise Octavie Baudenent Baroness Le Pelletier, needed to be disambiguated from Madeleine Pelletier, the French feminist and socialist who worked for the Red Cross during the war.

that provided a fuller picture of their significance, as seen in the <person> and <place> entries for the baron's example:

```
<person xml:id="baron-pelletier" n="pelletierl" corresp="#baroness-pelletier #silly #milon">
    <persName>Louis Henri Baron Le Pelletier</persName>
    <relation>Acquaintance</relation>
    <desc>
        <p>
            Breckinridge mentions that she has lunched with "a French baron and his wife" who
            have been benevolent to their village, which is located near La Ferté-Milon.
            Breckinridge indicates that the affective and material support that the couple have
            provided for their community is a part of their political agenda as "extreme
            radicals" who oppose the "reactionary" government in France (i.e., the postwar Bloc
            National). Described as a "radical socialist with excellent manners," the Baron
            mentioned in the letter of 1 May 1919 is likely the same mentioned (and named) in
            the letter of 31 August 1919. Baron Pelletier, the scion of a long-established
            family in the French nobility, inherited an extensive library from his father Louis
            Ernest and became known as an archivist, paleographer, and antiquarian associated
            with the <foreign xml:lang="fr">Bibliothèque de l'Arsenal</foreign> in Paris (<ref
            target="#valadon">Valadon 183</ref>; <ref target="#shc">SHC 28</ref>).
        </p>
    </desc>
</person>

<place xml:id="milon" n="fertemilon" corresp="#baron-pelletier #baronness-pelletier #silly">
    <placeName>La Ferté-Milon</placeName>
    <region type="department">Aisne</region>
    <region type="region">Hauts-de-France</region>
    <country>France</country>
    <location>
        <geo>49.1779 3.1247</geo>
    </location>
    <desc>
        <p>
            Breckinridge mentions that she has lunched with a French Baron and his wife (Louis
            Henri Baron Le Pelletier and Louise Octavie Baudenent Baroness Le Pelletier) who
            live in and are good to their village near La Ferté-Milon. The couple, Breckinridge
            indicates, are "extreme socialists" in opposition to the current "reactionary"
            government of France, i.e., the postwar Bloc National.
        </p>
    </desc>
</place>
```

These header entries were placed in the <teiHeader> element at the beginning or top of the XML corpus file outside of any individual student's letter, each of which occupied a <TEI> element.

Students were therefore able to collaborate on entries for named entities that appear in multiple documents for a more sophisticated understanding of their significance in Breckinridge's letters and life during her first year in the Aisne.

In addition to tagging and developing header entries for named entities, students added their own editorial notes following Matthias Bauer and Angelika Zirker's work on the theory and practices of literary annotation in the digital humanities. "What does annotation do to the text," and "[w]hat does it do for the reader," Bauer and Zirker ask as they explore how in practice we can "attempt to address the individual needs of readers while considering the kind and amount of information…they require in order to understand and interpret the text." As in the case of text encoding and the TEI Guidelines where there are usually more possibilities for markup than are useful for a project, Bauer and Zirker cue us to think about "where to stop when annotating a literary text." In particular, they worry that "the endless opportunity that the [digital] medium gives us to publish material is also its greatest weakness" given that readers will not be served well if left with a "glut" of superficial links to data residing in other locations such as websites, wikis, and encyclopedias. Echoing arguments for 'boutique' approaches to data, Bauer and Zirker favor the careful curation of 'in-house' annotations that evince a shared vision for the digital edition. While our project dealt in historical correspondence rather than literature per se, Bauer and Zirker's taxonomy—linguistic, formal, intratextual, intertextual, contextual, and interpretive annotations—was helpful in identifying moments in the letters that merited further explanation. Throughout this process students continued to collaborate on shared references and annotations that applied to multiple letters. Figure 21 below shows a facsimile detail from Breckinridge's letter to her mother dated 1 May 1919, followed by instructor-generated markup

for deletions, additions, editorial regularizations, line breaks, place entries, and editorial annotations. The subsequent XML begins about two-thirds through the paragraph in Figure 21.

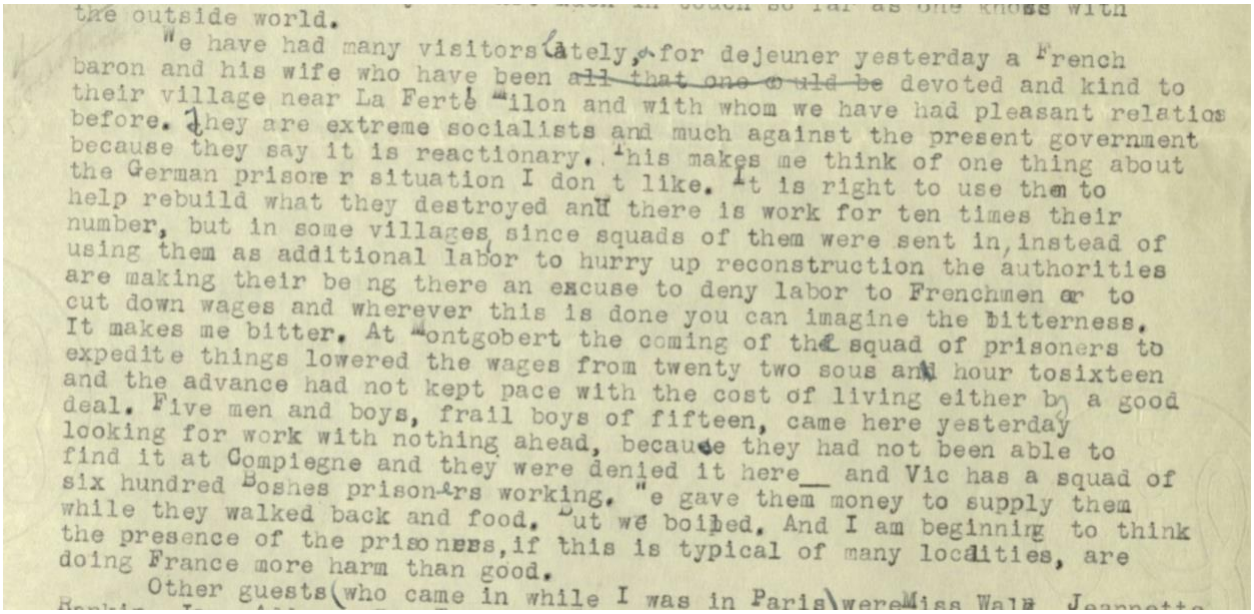

the outside world.
"e have had many visitors lately, for dejeuner yesterday a French baron and his wife who have been all that one could be devoted and kind to their village near La Ferté "ilon and with whom we have had pleasant relatios before. They are extreme socialists and much against the present government because they say it is reactionary. This makes me think of one thing about the German prisoner situation I don t like. It is right to use them to help rebuild what they destroyed and there is work for ten times their number, but in some villages since squads of them were sent in, instead of using them as additional labor to hurry up reconstruction the authorities are making their being there an excuse to deny labor to Frenchmen or to cut down wages and wherever this is done you can imagine the bitterness. It makes me bitter. At "ontgobert the coming of the squad of prisoners to expedite things lowered the wages from twenty two sous an hour tosixteen and the advance had not kept pace with the cost of living either by a good deal. Five men and boys, frail boys of fifteen, came here yesterday looking for work with nothing ahead, because they had not been able to find it at Compiegne and they were denied it here__ and Vic has a squad of six hundred Boshes prisoners working. "e gave them money to supply them while they walked back and food. But we boiled. And I am beginning to think the presence of the prisoners, if this is typical of many localities, are doing France more harm than good.
Other guests who came in while I was in Paris were Miss Walk Jeannette

Figure 21: detail from Mary Breckinridge's correspondence to her mother dated May 1, 1919 (courtesy of University of Kentucky Libraries Special Collections Research Center)

```
Five men and boys, frail boys of fifteen, came here yesterday <lb/>
looking for work with nothing ahead, becau
    <del rend="overwritten">e</del>
    <add quantity="1" unit="chars" place="over" rend="handwritten">s</add>
e they had not been able to <lb/>
find it at <placeName xml:id="compiegne">Compiegne</placeName> and they were denied it here
    <choice>
        <orig>_</orig>
        <reg resp="#studentname"/>
    </choice>
and <placeName xml:id="vsa">Vic</placeName> has a squad of <lb/>
six hundred Boshes<ptr type="annotation" target="#bosh-etymology"/>
    <note type="annotation" subtype="linguistic" resp="#studentname" xml:id="bosh-etymology">
        In the April-September 1916 volume of <title level="j">Current History: A Monthly
        Magazine of the New York Times</title>, Douglas Buffum confirms Breckinridge's
        observation that the term "Boshes" is "almost universally used" in France to refer to
        the Germans. "Boche," as Buffum spells it, is said to be "an abbreviation of <foreign
        xml:lang="fr">caboche</foreign>," which loosely signifies "a big, thick head."
        Originally used in France during the mid nineteenth-century to describe a
        "disagreeable, troublesome fellow," the term began to be used specifically against
        Germans assistants of Parisian printers as derogatory slang for their slowness in
        comprehending the intricacies of the French language. "The next step," Buffum
```

```
        indicates, was to apply the term to anyone of German origin <ref target="#buffum">
        (525)</ref>.
    </note>
prison
    <add rend="handwritten" place="inline">e</add>
rs working. We gave them money to supply them <lb/>
while they walked back and food. But we boi
    <del rend="overtyped">p</del>
    <add quantity="1" unit="chars" place="over" rend="typed">l</add>
ed. And I am beginnin
    <add quantity="1" unit="chars" place="over" rend="typed">g</add>
    <choice>
        <orig><space quantity="2" unit="chars"/></orig>
        <reg resp="#studentname"> </reg>
    </choice>
to think <lb/>
the presence of the prison
    <del rend="overtyped">rs</del>
    <add quantity="2" unit="chars" place="over" rend="typed">er</add>
s, if this is typical of many loca
    <add quantity="1" unit="chars" place="over" rend="typed">l</add>
ities, are <lb/>
doing France more harm than good.<ptr type="annotation" target="#pow-labor"/>
    <note type="annotation" subtype="historical" resp="#studentname" xml:id="pow-labor">
        In <title level="m">Violence Against Prisoners of War in the First World War</title>,
        Heather Jones writes that France, of all the Allied nations, made conspicuous and
        controversial use of German POW labor. The day after the Armistice, 100,000 German POW
        laborers were sent from the French interior to the devastated regions in the north not
        only to "work on reconstruction and de-mining projects" but also to "free up jobs in
        the interior for returning French servicemen" (an ironic goal considering
        Breckinridge's critique that the displaced POWs ended up taking French jobs in the
        north). Jones corroborates Breckinridge's pathos-laden anecdote of the hungry French
        vagabonds seeking work; "several prefects in the devastated areas," Jones reports,
        protested the influx of German POWs precisely for their impact on the region's ability
        to feed its people. Though Breckinridge voices her anger at the German soldiers (which,
        writ large, contributed to the retention and exploitation of German POW labor), she
        recognizes the harmful effect of their prolonged presence, which had increased to
        270,000 in the region by the time of this letter <ref target="#jones-pows">(296-97)
        </ref>.
    </note>
```

"Textual encoding has never been as sexy as text analysis," writes Ryan Cordell; yet, it involves an intentional effort towards transparency and public accountability regarding "the relationships among preservation, presentation, access, and interpretation" ("On Ignoring Encoding"). Students indeed found text encoding to be demanding and detail-intensive but they

also “found even the most granular acts of description to be profoundly interpretive,” blurring the line between technical and intellectual labor (Conatser et al. 30). Periodically dedicating class time to brief case studies reinforced the modeling work that continued throughout the course project and connected the minutiae of markup to significant ethical and scholarly questions. Two examples that serve well in this role concern the encoding of racialized eye dialect and personal pronouns. In the first case, editors of *Race and Children’s Literature in the Gilded Age* sought to regularize eye dialect for machine readability without making claims of value or correctness in their handling of the markup (Gailey). While I discuss this example in detail in chapter one, it hinges on how `<sic>` and `<corr>` convey editorial arguments (in this case, about text portraying racialized speech). Students may consider how *sic* as an editorial term often appears in texts to indicate errors of grammar, spelling, or syntax; to call attention to the non-standard or deviant nature of the text; and perhaps even to ensure that readers understand that the original author(s) and not the editor(s) are responsible for the so-called error. They can discuss how the correction element `<corr>` may reinforce the notion of erasing a dialect so that a text better aligns with the sensibilities of a particular imaged readership. Most importantly, students explore questions such as: why encode dialect in the first place? what would be the purposes, benefits, and drawbacks? how does the complicated authorship and cultural legacy of the case study text—the Uncle Remus stories of the late nineteenth-century written by white folklorist Joel Chandler Harris and inspiration for the now-excoriated 1946 Disney film *Song of the South*—affect our thinking in response to these questions? considering the solution that the editors of *Race and Children’s Literature in the Gilded Age* adopted (i.e., `<orig>` and `<reg>`), are there alternatives for encoding eye dialect in the TEI Guidelines either within or beyond the framework imposed by a `<choice>`

element? As a generative discussion, students might consider how they would handle eye dialect if they were to encounter it in any of the documents for their own project.

In the second case, a proposal initiated by Ashley M. Clark led to the formal adoption of markup for personal pronouns in the TEI Guidelines. Clark's original blog-style proposal on the TEI's GitHub site serves as a clear example of the thinking that goes into any modification of the TEI Guidelines, especially with the rich comments section including exchanges among the TEI Board, Technical Council, and larger community. Stating one's personal or preferred pronouns has grown into a normalized aspect of communication, Clark argues, with both transgender and cisgender writers adopting the practice, for example, in email signatures. Finding that the extant element `<sex>` is inadequate for containing information about pronouns, perhaps as an additional attribute such as `@persPronouns` for `<sex>`, Clark proposes a `<persPronouns>` element that would require both `@value` and `@evidence` attributes, the latter importantly making transparent how an editor has determined the pronouns for the person at hand. Additionally, she reveals, combining the computationally tractable attributes with qualitative parsed character data (i.e., the text within XML tags) allows for a "queer solution" of providing "the ability to pull statistics" via attribute values, "but also the ability to describe the subject's lived reality."

```
<persPronouns value="she/her/hers" evidence="self-identification">
    Prefers she/her/hers. Comfortable with they/them/theirs.
</persPronouns>
```

Prompted by Clark's example, students may consider other aspects of personal identity that may be important for a text encoding project, and how those aspects could be encoded in the `<person>` list in the `<teiHeader>`. These speculations pose both technical and editorial challenges regarding how potential additions or emendations would fit within the overall TEI framework as well as a

project's or audience's needs. Students may consider approaches to these and other issues and compare their ideas with the solutions advanced by scholar-editors for real projects.

<Transformation and Sustainability/>

One of the fundamental concepts of digital scholarly editing is the distinction between encoding and presentation: "to record and document the physical, structural, and semantic data" on the one hand and "to determine and instruct how the registered data of the source material are to be processed with regard to selection, display, and format" on the other (Pichler and Bruvik 180-81). Students recognized this distinction as critical to the sustainability of our digital edition as they produced a platform-independent corpus file of descriptive markup that could be adapted for any number of venues or research projects through XSLT or other languages such as XQuery. A single semester did not provide time for students to learn other markup and web development languages, i.e., not only the XSLT but also the HTML, CSS, and JavaScript that are required to generate a browser-based electronic edition. In place of the actual coding, students addressed the production of digital editions from conceptual and design standpoints. Midway through the semester they presented and led discussions on published digital documentary editions with an eye towards our own. Students plotted out structure, content, and user experience given that digital documentary editions record "as many features of the original document as are considered meaningful by the editors, displayed in all the ways the editors consider useful for the readers, including all the tools necessary to achieve such a purpose" (Pierazzo 475). Figure 22 below shows an example of one feature of the original documents and one display and tool that the

class decided would be useful for readers: a collection of the places mentioned in the letters associated with our editorial summaries of their significance, displayed in Google Maps so that readers may explore the spatial dimension of the experiences recounted in the correspondence.

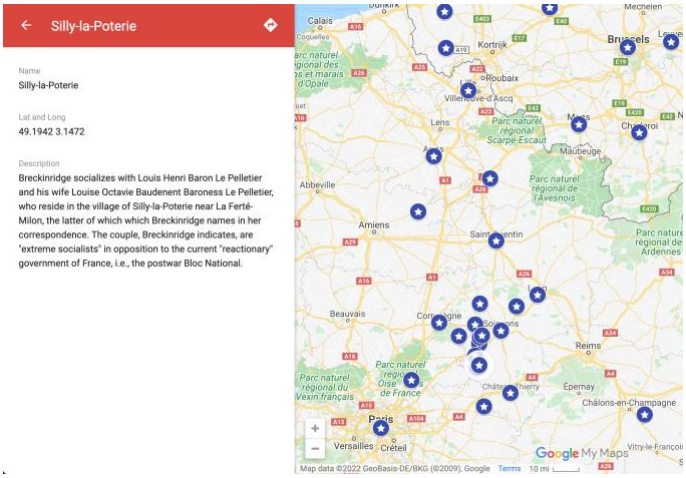


Figure 22: detail of a Google map with locations and information about places mentioned in Breckinridge's letters

In addition to Elena Pierazzo's rationale for digital documentary editions, students consulted the Association for Documentary Editing's standards for electronic editions, the Modern Language Association's guidelines for editors of scholarly editions, and the University of Pittsburgh Visual Media Workshop's socio-technical sustainability roadmap.[127] The learning goals for this phase of the course focused on understanding and appreciating considerations for future development; were the curriculum year-long, it could involve students directly in designing, transforming, and even preparing the edition for scholarly review. Broadly, I led students to understand scholarly editing itself as another kind of modeling, with *digital* scholarly editing prompting us to consider

[127] ADE, "Minimum Standards for Electronic Editions," www.documentaryediting.org/wordpress/?page_id=508; MLA, "Guidelines for Editors of Scholarly Editions," www.mla.org/Resources/Research/Surveys-Reports-and-Other-Documents/Publishing-and-Scholarship/Guidelines-for-Editors-of-Scholarly-Editions; The Visual Media Workshop, "The Socio-Technical Sustainability Roadmap, sites.haa.pitt.edu/sustainabilityroadmap/.

a different ethos for the published scholarly edition than what we've become accustomed to in print. Drawing from Martha Nell Smith, executive editor of the *Dickinson Electronic Archives*, we see how this ethos embraces "diverging views" about the work(s), "[m]ultiple authorities" for analysis and interpretation, exigent issues and interests as opposed to the authoritative or correct edition, disagreements among editors, documentation and transparency of editorial decisions and processes, and "the 'messy' concerns of diverse humanity," which, Smith emphasizes contra the trends of the field, "matter for bibliographic and electronic encoding and presentation" (14).

A critical part of the course's public engagement and students' reflection on their learning was a group presentation for local stakeholders on campus and a collaboratively written essay for a specific venue and audience. We devoted class time during the final month of the semester to plan, develop, and finalize both of these assignments. I advertised the presentation across campus and particularly to audiences for whom our project would resonate because of a connection with the edition's subject matter or with our archival and digital methods. The pedagogy of assigning and leading students through a collaborative essay deserves its own chapter, but I will note here that it requires proactive and creative thinking on the instructor's part to secure an authentic venue for student work. For one instance of the course, for example, I explored a peer-reviewed medical humanities blog before turning to an archival organization's newsletter. Writing for a real audience, venue, and purpose fulfilled the course's promise that students would be taken seriously as editors, and that their work would matter as a form of public scholarship. It would, as we found, underscore the public stakes of our collective learning and clarify what it means, à la Kirschenbaum, for pedagogy to be open and visible, bound in infrastructures both archival and digital, and deeply committed to collaboration.

With crises of public health, climate change, social injustice, and political instability, the early 2020s strike teacher-scholars as a reckoning not only with the orthodoxies of higher education but also with how we have traditionally imagined innovation in teaching and learning. In one take on what this moment 'means' for academics, teaching and learning author and center director Paul Hanstedt urges us to engage students in issues of public interest that "foreground [the] messy interactions" between disciplines and professions, such as "how data influences poetry and how poetry can shape our ability to give meaning to data." Moreover, he argues, educators are called on to explore the "possibility of agency" for students in more intentional and systematic ways. Now a decade beyond the 'digital humanities moment' that anticipated a more publicly engaged and publicly visible scholarly community, our current moment asks us to imagine our students as part of that community and to *re*imagine their learning through open pedagogy and public-facing work. The approach I've described in this chapter is one way to respond to that call, with students exploring how text encoding and digital scholarly editing influence our understanding of historical records and how those records grant palpable meaning and public stakes to digital humanities learning environments. Code, in general, provides a means to (re)examine complex and emerging phenomena all the while representing one of those phenomena itself. We will do well to continue to reckon with code in the humanities not just as a research interest but as a pedagogical prompt to engage students in ways that match the nuance that we seek when teaching about language, writing, rhetoric, composition, and communication.

Afterword

In the same year that this dissertation was completed, the publication of the open-access collection *Critically Conscious Computing* represents a significant step in imagining how code has a place in disciplines and learning environments outside of traditional computer science and programming classes. More than this, the collection aims to guide educators towards practicing a critical pedagogy that foregrounds the symbolic, rhetorical, and cultural work of code beyond its operation as a purely instrumental system of instructions and operations. Drawing explicitly from the Freirean tradition, Amy J. Ko describes the pedagogy that drives the collection as one that "question[s] the application of [computing] power, ask[s] who has that power, ponder[s] how that power should be distributed, and insist[s] on the responsibility of those who possess it" ("Critical CS"). This pedagogical commitment manifests, for example, in explorations of how educators can lead students to critique how information is encoded for computational purposes, particularly as they understand encoding as a *transformative* act rather than a transcriptional one. "Bits can model anything, but not without consequences," warns the epigraph to the chapter that takes encoding as its focus (Ko, "Encoding Information"). While the collection is intended for secondary educators, it prompts those of us in postsecondary education to consider how we can extend a critical consciousness around code into our own teaching so that students can continue to see it in new and empowering ways. This dissertation is precisely such an effort, and I have sought to expand how we conceptualize and harness code as a literacy and medium for writing and action—even a way of making sense of the world. If "[c]ode is a story we tell about data,"

as Paul Ford writes, I would add that it's also a story we tell about, to, and with each other ("Real Programming").

And, as all stories do, it continues to evolve. In November 2022, the *New York Times* reported on a lawsuit seeking class action status against Microsoft, Github, and OpenAI. The litigation concerns Copilot, "a new kind of artificial intelligence technology that could generate its own computer code" (Metz). In the same way that text composing platforms (such as the Microsoft Outlook email client) suggest the next word or phrase based on what is currently being written, Copilot has been trained on vast amounts of code publicly available on the web so that it might allow programmers to insert lines or even entire sections of predictively generated code in the name of making their labor more efficient and focusing their attention on the more creative or complex issues at hand. According to the *Times* reporting, at the heart of the lawsuit are "the legal rights of millions of programmers who spent years writing the original code" that Copilot uses to train itself to make better suggestions and generate more useful code, thereby increasing its market value while potentially encroaching upon the open-source programming communities that usually serve as hubs for the exchange of code and ideas (Metz). Viewed through the lenses of digital humanities, rhetoric, and composition studies, the Copilot case inspires new questions about code as a form of writing and communication. Beyond issues around intellectual property, what does it mean for code to be 'written' by a non-human agent? Given what I've attempted to establish in this dissertation—for example, the rhetoricity of code—what is at stake in the project of automating codewriting based on available corpora of codetexts? In what ways does our sense of coding literacy incorporate the negotiation of text and meaning with an elusive agent among the networks of Copilot's deep learning processes and the seemingly incoherent aggregate of its

source data? Is all of this any different than <oXygen/> automatically adding an identical closing tag when we open a new element for the sake of well-formed markup? For the part of humanities teacher-scholars, the story must continue to center its learner-protagonists as they become critical and imaginative users who engage these sorts of questions as deeply human ones that exceed but also profoundly implicate the more technical areas from which they seem to emerge.

~

Readers of this dissertation may hear in its title an echo of Stephen A. Bernhardt's 1986 article "Seeing the Text" in *College Composition and Communication*. Bernhardt's goal is, of course, quite different than mine; emphasizing "[t]he physical fact of the text," he urges fellow compositionists to heed what we now would describe as document and information design: e.g., how nondiscursive signifiers such as arrangement, spacing, stylization, iconography, and color reveal "the structure or logic" of a visually driven text in ways that aid and make more efficient a reader's understanding of a text's message (66, 68). In a way, though, my project shares with Bernhardt an inquiry around where meaning lies, not only for our ability to be better readers of a text, but to compose more persuasively, effectively, and ethically for new audiences and in new contexts. I have deliberately courted slippage between notions of code and text, and I have also devoted much of my analysis to code's revelatory capacity for the structure or logic that it both draws from and creates. Seeing the code, especially amid the seas of code that increasingly affect our lived experiences, bears value and urgency for humanities education not only in the ways I've investigated here but also for those unknowns, a la McGann, that we have yet to imagine.

Works Cited

@amclark42. “TEI just got a little more inclusive and a little more queer: <persPronouns> has arrived!!!” *Twitter*, 25 Feb. 2021, 5:38 p.m., twitter.com/amclark42/status/1365068593700954112.

@MikeBloomberg. “My New Year's resolution is to learn to code with Codecademy in 2012! Join me. http://codeyear.com/ #codeyear.” *Twitter*, 5 Jan. 2012, 10:56 a.m., twitter.com/mikebloomberg/status/154999795159805952.

@quinnanya. “Hats off to @TEIconsortium for an update with major improvements to #MultilingualDH support (Ruby annotations) and Queer DH support (personal pronoun element)! [Party popper] (h/t @amclark42)” *Twitter*, 25. Feb 2021, 10:26 p.m., twitter.com/quinnanya/status/1365141248093298688.

Abbate, Janet and William Aspray. *Recoding Gender: Women’s Changing Participation in Computing*. MIT Press, 2012.

Abel, Jaison R. and Richard Deitz. “Agglomeration and job matching among college graduates.” *Regional Science and Urban Economics*, no. 51, 2015, pp. 14-24. https://doi.org/10.1016/j.regsciurbeco.2014.12.001.

Abelson, Harold and Gerald Jay Sussman. *Structure and Interpretation of Computer Programs* 2nd edition. MIT Press, 1996, mitpress.mit.edu/sites/default/files/sicp/full-text/book/book-Z-H-1.html.

Acker, Amelia. “Metadata.” In *Uncertain Archives: Critical Keywords for Big Data*, edited by Nanna Bonde Thylstrup, Daniela Agostinho, Annie Ring, Catherine D’Ignazio, and Kristen Veel, MIT Press, 2021, pp. 321-329.

Ackermans, Hannah. “How an Academic Companion Website Makes Media-Specific Arguments.” *American Quarterly*, vol. 72, no. 4, 2020, pp. 1011-1020.

Anderson, Erin R. and Trisha N. Campbell. “Ethics in the Making.” In *Making Things and Drawing Boundaries: Experiments in the Digital Humanities*, edited by Jentery Sayers, University of Minnesota Press, 2017, dhdebates.gc.cuny.edu/read/untitled-aa1769f2-6c55-485a-81af-ea82cce86966/section/2bac4a0c-e91c-4873-801c-c113190700fb.

Applen, J. D. “Technical Communication, Knowledge Management, and XML.” *Technical Communication*, vol. 49, no. 3, 2002, pp. 301-313.

Ball, Cheryl. “Logging On.” *Kairos: A Journal of Rhetoric, Technology, and Pedagogy*, vol. 17, no. 2, 2013, kairos.technorhetoric.net/17.2/loggingon/index.html.

Ball, Cheryl, Tarez Samra Graban, and Michelle Sidler. “The Boutique is Open: Data for Writing Studies.” Personal Website, 2013, ceball.com/wp-content/uploads/2013/11/NHUK-chapter-rhetoric.io-PREPRINT.pdf.

Bangert-Drowns, Robert L. “The Word Processor as an Instructional Tool: A Meta-Analysis of Word Processing in Writing Instruction.” *Review of Educational Research*, vol. 63, no. 1, Spring 1993, pp. 69-93.

Bass, Randall. “What’s the Problem Now?” *To Improve the Academy: A Journal of Educational Development*, vol. 39, no. 1, 2020, dx.doi.org/10.3998/tia.17063888.0039.102.

Bauer, Matthias and Angelika Zirker. “Whipping Boys Explained: Literary annotation and digital humanities.” In *Literary Studies in the Digital Age: An Evolving Anthology*, edited by Kenneth M. Price and Ray Siemens, Modern Language Association, 2015, dlsanthology. mla.hcommons.org/whipping-boys-explained-literary-annotation-and-digital-humanities/.

Beam, Carey and Carrie Schwier. “Learning in Place: The Teaching Archivist and Place-Based Education.” *Archival Issues: The Journal of the Midwest Archives Conference*, vol. 29, no. 1, 2018, pp. 7-25.

Beaufort, Anne. *College Writing and Beyond: A New Framework for University Writing Instruction*. Utah State University Press, 2007.

Beck, Estee. “A Theory of Persuasive Computer Algorithms for Rhetorical Code Studies.” *Enculturation: A Journal of Rhetoric, Writing, and Culture*, 23, 2016, enculturation.net/a-Theory-of-persuasive-computer-algorithms.

Behrent, Michael C. “Foucault and Technology.” *History and Technology: An Interactive Journal*, vol. 29, no. 1, 2013, pp. 54-104.

Bender, Emily, Timnit Gebru, Angelina McMillan-Major, and Margaret Mitchell. “On the Dangers of Stochastic Parrots: Can Language Models Be Too Big?” *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, March 2021, pp. 610-623. Accessed 23 Oct. 2021 from dl.acm.org/doi/10.1145/3442188.3445922.

Benjamin, Walter. “The Work of Art in the Age of Mechanical Reproduction.” In *Illuminations*, edited by Hannah Arendt, translated by Harry Zohn, Shocken Books, 1968, pp. 217-252.

Bernhardt, Stephen A. “Seeing the Text.” *College Composition and Communication*, vol. 37, no. 1, Feb. 1986, pp. 66-78.

Berry, David. “Critical Digital Humanities.” *Conditio Humana*, conditiohumana.io/critical-digital-humanities/. Accessed 13 December 2020.

Berry, David M., Beatrice Fazi, Ben Roberts, and Alban Webb. “No Signal without Symbol: Decoding the Digital Humanities.” In *Debates in the Digital Humanities 2019*, edited by Matthew K. Gold and Lauren F. Klein, University of Minnesota Press, 2019, dhdebates.gc.cuny.edu/read/untitled-f2acf72c-a469-49d8-be35-67f9ac1e3a60/section/687d2d32-0ec6-4158-b776-ee5cb989ef8b#ch05.

Bjork, Olin. “Digital Humanities and the First-Year Writing Course.” In *Digital Humanities Pedagogy: Practices, Principles, and Politics*, edited by Brett D. Hirsch, Open Book Publishers, 2012, pp. 97-119.

Blythe, Stuart, Claire Lauer, and Paul G. Curran. “Professional and Technical Communication in a Web 2.0 World.” *Technical Communication Quarterly*, vol. 23, no. 4, 2014, pp. 265-287.

Boggs, Jeremy, Jennifer Reed, and J. K. Purdom Lindblad. “Making It Matter.” In *Making Things and Drawing Boundaries: Experiments in the Digital Humanities*, edited by Jentery Sayers, University of Minnesota Press, 2017, dhdebates.gc.cuny.edu/read/untitled-aa1769f2-6c55-485a-81af-ea82cce86966/section/f29e68c9-e28e-4c1a-b083-93025a5ed6b3#ch37.

Bogost, Ian. *Alien Phenomenology, or What It’s Like to Be a Thing*. University of Minnesota Press, 2012.

———. *Persuasive Games: The Expressive Power of Videogames*. MIT Press, 2010.

———. “The Cathedral of Computation.” *The Atlantic*, 15 Jan. 2015, www.theatlantic.com/technology/archive/2015/01/the-cathedral-of-computation/384300/

Bork, John. *Critical Programming: Toward a Philosophy of Computing*. 2015. University of Central Florida, PhD dissertation.

Box, George E. P. and Norman R. Draper. *Empirical Model-Building and Response Surfaces*. Wiley, 1987.

“brainfuck.” *Esolangs Wiki*, esolangs.org/wiki/Brainfuck. Accessed 30 April 2022.

Brandt, Deborah. *Literacy in American Lives*. Cambridge University Press, 2001.

Brannock, Jennifer, Craig Carey, and Joyce O. Inman. “Starting from the Archives: Digital Humanities Partnerships, Projects, and Pedagogies.” In *Digital Humanities, Libraries, and Partnerships: A Critical Examination of Labor, Networks, and Community*, edited by Robin Kear and Kate Joranson, Chandos Publishing, 2018, pp. 163-176.

Brock, Kevin. *Rhetorical Code Studies: Discovering Arguments In and Around Code*. University of Michigan Press, 2019.

Brooks, Kevin and Chris Lindgren. “Responding to the Coding Crisis: From Code Year to Computational Literacy.” In *Strategic Discourse: The Politics of (New) Literacy Crises*, edited by Lynn C. Lewis, Computers and Composition Digital Press/Utah State University Press, 2015, ccdigitalpress.org/book/strategic/chapters/brookslindgren/index.html.

Brownell, Sara E., Jordan V. Price, and Lawrence Steinman. “Science Communication to the General Public: Why We Need to Teach Undergraduate and Graduate Students this Skill as Part of Their Formal Scientific Training.” *The Journal of Undergraduate Neuroscience Education (JUNE)*, vol. 12, no. 1, 2013, pp. E6-E10.

Buehl, Jonathan, Tamar Chute, and Anne Fields. “Training in the Archives: Archival Research as Professional Development.” *College Composition and Communication*, vol. 64, no. 2, 2012, pp. 274–305.

Buffum, Douglas. “Origin of the Word ‘Boche.’ ” *Current History: A Monthly Magazine of the New York Times*, vol. 4, 1916, p. 525.

Buolamwini, Joy and Timnit Gebru. “Gender Shades: Intersectional Accuracy Disparities in Commercial Gender Classification.” *Proceedings of Machine Learning Research*, no. 81, pp. 1-15.

Burnard, Lou. “The Evolution of the Text Encoding Initiative: From Research Project to Research Infrastructure.” *Journal of the Text Encoding Initiative*, no. 5, 2013, journals.openedition.org/jtei/811.

Byrd, Antonio. “Between Learning and Opportunity: A Study of African American Coders’ Networks of Support.” *Literacy in Composition Studies*, vol. 7, no. 2, pp. 31-55, licsjournal.org/index.php/LiCS/article/view/708.

Callaway, Elizabeth, Jeffrey Turner, Heather Stone, and Adam Halstrom. “The Push and Pull of Digital Humanities: Topic Modeling the ‘What is digital humanities?’ Genre.” *Digital Humanities Quarterly*, vol. 14, no. 1, 2020, www.digitalhumanities.org/dhq/vol/14/1/000450/000450.html.

Cangialosi, Karen. “But you can’t do that in a STEM course!” *Hybrid Pedagogy*, 26 June 2018, hybridpedagogy.org/do-in-a-stem-course/.

Carlson, Scott. “What Gets Forgotten in Debates About the Liberal Arts.” *The Chronicle of Higher Education*, 05 March 2018, www.chronicle.com/article/What-Gets-Forgotten-in-Debates/242749.

CAST. “Provide Options for Executive Functions.” *Universal Design for Learning Guidelines* version 2.2, 2018, udlguidelines.cast.org/action-expression/executive-functions/executive-functions.

Caton, Paul. “Pure Transcriptional Markup” Abstract. *The Annual International Conference of the Alliance of Digital Humanities Organizations, 16-19 July 2013*, dh2013.unl.edu/abstracts/ab-245.html.

Cayley, John. “The Code is not the Text (Unless It Is the Text).” *Electronic Book Review*, 10 Sep. 2002, electronicbookreview.com/essay/the-code-is-not-the-text-unless-it-is-the-text/.

Cecire, Natalie. “When Digital Humanities Was in Vogue.” *Journal of Digital Humanities,* vol. 1, no. 1, Winter 2011, journalofdigitalhumanities.org/1-1/when-digital-humanities-was-in-vogue-by-natalia-cecire/.

Chun, Wendy Hui Kyong. “The Enduring Ephemeral, or the Future Is a Memory.” *Critical Inquiry*, vol. 35, no. 1, 2008, pp. 148-171.

———. *Programmed Visions: Software and Memory*. The MIT Press, 2011.

Clark, Ashley M. 2020. “Proposal: new element <persPronouns> #2010.” Text Encoding Initiative Repository (GitHub). June 29, 2020. github.com/TEIC/TEI/issues/2010.

Coley, Toby. *Teaching with Digital Media in Writing Studies: An Exploration of Ethical Responsibilities*. Peter Lang, 2012.

Comer, Kathryn B. and Michael Harker. “The Pedagogy of the Digital Archive of Literacy Narratives: A Survey.” *Computers and Composition: An International Journal*, vol. 35, 2015, pp. 65-85.

Conatser, Trey. “Changing Medium, Transforming Composition.” *Journal of Digital Humanities*, vol. 2, no. 2, 2013, journalofdigitalhumanities.org/2-2/changing-medium-transforming- composition-by-trey-conatser/.

———. Syllabus for First-Year Writing: Codes. Department of English, The Ohio State University, Columbus, Spring 2013.

Conatser, Trey, et al. “Coding (and) the Archive: Texts, Markup, and Open Pedagogy.” *Midwest Archives Conference Newsletter*, vol. 47, no. 1, 2019, pp. 29-31.

Conger, Kate. “Exclusive: Here's The Full 10-Page Anti-Diversity Screed Circulating Internally at Google [Updated].” *Gizmodo*, 5 Aug. 2017, gizmodo.com/exclusive-heres-the-full-10-page-anti-diversity-screed-1797564320.

Cordell, Ryan. “How Not to Teach Digital Humanities.” In *Debates in the Digital Humanities 2016*, edited by Lauren F. Klein and Matthew K. Gold, University of Minnesota Press, 2016, dhdebates.gc.cuny.edu/read/untitled/section/31326090-9c70-4c0a-b2b7-74361582977e.

———. “On Ignoring Encoding.” Personal Website. May 8, 2014. ryancordell.org/research/dh/on-ignoring-encoding/.

correspSearch: Search Scholarly Editions of Letters. Correspsearch.net. Accessed 10 March 2019.

Cottom, Tressie McMillan. “More Scale, More Questions: Observations from Sociology.” In *Debates in the Digital Humanities 2016*, edited by Lauren F. Klein and Matthew K. Gold, University of Minnesota Press, 2016, dhdebates.gc.cuny.edu/read/untitled/section/55e48b34-543a-41f7-97c9-8c8643bf8844#ch47.

———. “Rethinking the Context of Edtech.” *Educause Review*, vol. 54, no. 3, 2019, pp. 22-29, er.educause.edu/-/media/files/articles/2019/8/er193103.pdf.

Council of Writing Program Administrators. “WPA Outcomes Statement for First-Year Composition (3.0).” 17 July 2014 (approved), 18 July 2019 (revised), wpacouncil.org/aws/CWPA/pt/sd/news_article/243055/_PARENT/layout_details/false.

Cox, Geoff and Alex McLean. *Speaking Code: Coding as Aesthetic and Political Expression*. MIT Press, 2012.

Cox, Geoff, Alex McLean, and Adrian Ward. “Coding Praxis: Reconsidering the Aesthetics of Code.” In *read_me: Software Art & Cultures*, edited by Olga Goriunova and Alexei Shulgin, Digital Aesthetics Research Centre, University of Aarhus, 2004, pp. 161-174.

Crawford, Cole. *Respect the Gap: From Big to Boutique Data through* Laboring-Class Poets Online. 2017. Oregon State University, MA Thesis.

Croft, James, Phyllis Conn, Joseph Serafin, and Rebecca Wiseheart. “Attempting to Connect Disciplinary Principles of ‘Effective Writing’ with Students’ Prior Writing Experiences in Four Disciplines.” In *Diverse Approaches to Teaching, Learning, and Writing Across the Curriculum: IWAC at 25*, edited by Lesley Erin Bartlett, Sandra L. Tarabochia,

Andrea R. Olinger, and Margaret J. Marshall, WAC Clearinghouse, 2020, pp. 193-208.

Cummings, James. "A World of Difference: Myths and Misconceptions about the TEI." *Digital Scholarship in the Humanities*, vol. 34, supp. 1, 2019, pp. 58-79.

Dastin, Jeffrey. "Amazon scraps secret AI recruiting tool that shows bias against women." *Reuters*, 9 October 2018, www.reuters.com/article/us-amazon-com-jobs-automation-insight/amazon-scraps-secret-ai-recruiting-tool-that-showed-bias-against-women-id USKCN1MK08G. Accessed 25 March 2019.

Decker, Juilee. "No More 'Dusty Archive' Kitten Deaths: Discoverability, Incidental Learning, and Digital Humanities." In *Quick Hits for Teaching with Digital Humanities: Successful Strategies from Award-Winning Teachers*, edited by Christopher J. Young, Michael Morrone, Thomas C. Wilson, and Emma Annette Wilson, Indiana University Press, 2020, pp. 232-240.

"Digital Humanities 2011: June 19-22." *Stanford University*, dh2011.stanford.edu/?page_id=97. Accessed 22 Jan 2022.

DeRosa, Robin and Rajiv Jhangiani. "Open Pedagogy." In *A Guide to Making Open Textbooks with Students, edited by Elizabeth Mays*, Rebus Community, 2017, press.rebus. community/makingopentextbookswithstudents/chapter/open-pedagogy/.

DiSessa, Andrea A. *Changing Minds: Computers, Learning, and Literacy*. MIT Press, 2001.

Dressel, Julia and Hany Farid. "The Accuracy, Fairness, and Limits of Predicting Recidivism." *Science Advances*, vol. 17, no. 1, 2018, advances.sciencemag.org/content/4/1/eaao5580.

Drucker, Johanna. "Humanities Approaches to Graphical Display." *Digital Humanities Quarterly*, vol. 5, no. 1, 2011, www.digitalhumanities.org/dhq/vol/5/1/000091/000091. Html.

Duke, Rachel C. S. and Sarah Stanely. " 'Decoding' with Encoding: Digital Tools in the Special Collections Classroom." In *Teaching Undergraduates with Archives*, edited by Nancy Bartlett, Elizabeth Gadelha, and Cinda Nofziger, Maize Books, 2019, pp. 60-70.

Dumont, Stefan, et al. "Correspondence Metadata Interchange Format (CMIF)." In *Encoding Correspondence: A Manual for Encoding Letters and Postcards in TEI-XML and DTABf*, edited by Stefan Dumont, Susanne Haaf, and Sabine Seifert. Berlin-Brandenburg Academy of Sciences and Humanities, 2019. encoding-correspondence.bbaw.de/v1/ CMIF.html.

Dyson, Anne Haas and Celia Genishi. *On the Case: Approaches to Language and Literacy Research.* Teachers College Press, 2005.

Easter, Brandee. " 'Feminist_brevity_in_light_of_masculine_long-windedness:' Code, Space, and Online Misogyny." *Feminist Media Studies,* vol. 18, no. 4, 2018, pp. 675-685.

———. "Fully Human, Fully Machine: Rhetorics of Digital Disembodiment in Programming." *Rhetoric Review*, vol 39, no. 2, 2020, pp. 202-215.

Earhart, Amy. "Can Information Be Unfettered? Race and the New Digital Humanities Canon." In *Debates in the Digital Humanities*, edited by Matthew K. Gold, University of Minnesota Press, 2012, dhdebates.gc.cuny.edu/read/untitled-88c11800-9446-469b-a3be-3fdb36bfbd1e/section/cf0af04d-73e3-4738-98d9-74c1ae3534e5.

Egginton, William. "Is Computer Code a Foreign Language?" *The New York Times*, 17 March 2019, www.nytimes.com/2019/03/17/opinion/code-foreign-language.html. Accessed 24 March 2019.

Elwood, Sarah and Matthew Wilson. "Critical GIS Beyond 'Week 10: Ethics.' " *International Journal of Geographical Information Science*, vol. 31, no. 2, 2017, pp. 1-19.

Engel, Deena and Marion Thain. "Textual Artifacts and their Digital Representations: Teaching Graduate Students to Build Online Archives." *Digital Humanities Quarterly*, vol. 9, no. 1, 2015, www.digitalhumanities.org/dhq/vol/9/1/000199/000199.html.

Ennis. Michael. "Explaining a Scientific Concept for Page and Screens." *Prompt: A Journal of Writing Assignments*, vol. 1, no. 1, 2017, pp. 47-54. thepromptjournal.com/index.php/prompt/article/view/14.

Enoch, Jessica and Pamela VanHaitsma. "Archival Literacy: Reading the Rhetoric of Digital Archives in the Undergraduate Classroom." *College Composition and Communication*, vol. 67, no. 2, 2015, pp. 216–242.

FairCopy. Performant Software Solutions LLC, 2020, www.faircopyeditor.com/en/. Accessed 18 December 2021.

"First-Year Writing Program Guidebook University of Georgia 2021-2022." www.english.uga.edu/sites/default/files/inline-files/FYW%20handbook%2021-22%20 full.pdf

Flanders, Julia. "Building Otherwise." In *Bodies of Information: Intersectional Feminism and Digital Humanities*, edited by Elizabeth Losh and Jacqueline Wernimont, University of Minnesota Press, 2018, dhdebates.gc.cuny.edu/read/untitled-4e08b137-aec5-49a4-83c0-38258425f145/section/f627035f-5fd0-4bd6-ad74-361374ed9a2a#ch16.

Flanders, Julie, Syd Bauman, and Sarah Connell. "Text Encoding." In *Doing Digital Humanities*, edited by Constance Compton, Richard J. Lane, and Ray Siemens, Routledge, 2016, pp.

104-122.

———. “XSLT: Transforming Our XML Data.” In *Doing Digital Humanities*, edited by Constance Compton, Richard J. Lane, and Ray Siemens, Routledge, 2016, pp. 255-272.

Flavell, John H. “Metacognition and Cognitive Monitoring: A New Area of Cognitive-Developmental Inquiry.” *American Psychologist*, vol. 34, no. 10, 1979, pp. 906-911.

Ford, Paul. “ ‘Real’ Programming Is an Elitist Myth.” *Wired*, 18 Aug. 2020, www.wired.com/story/databases-coding-real-programming-myth/. Accessed 30 November 2022.

———. “What Is Code?” *Businessweek*, 11 June 2015, www.bloomberg.com/graphics/2015-Paul-ford-what-is-code. Accessed 10 March 2019.

Framework for Success in Postsecondary Writing. Council of Writing Program Administrators, National Council of Teachers of English, and the National Writing Project, 2011, wpacouncil.org/aws/CWPA/asset_manager/get_file/350201?ver=7548.

Fyfe, Paul. “Digital Pedagogy Unplugged.” *Digital Humanities Quarterly*, vol. 5, no. 3, 2011, digitalhumanities.org/dhq/vol/5/3/000106/000106.html.

Gabourey, Jacob. “Critical Unmaking: Toward a Queer Computation.” In *The Routledge Companion to Media Studies and Digital Humanities*, edited by Jentery Sayers, Routledge, 2018, pp. 483-491.

Gailey, Amanda. “A Case for Heavy Editing: The Example of *Race and Children’s Literature in the Gilded Age*.” In *The American Literature Scholar in the Digital Age*, edited by Amy E. Earhart and Andrew Jewell. University of Michigan Press, 2011, pp. 125-144.

Gilliard, Chris. “Pedagogy and the Logic of Platforms.” *Educause Review*, 3 July 2017, er.educause.edu/articles/2017/7/pedagogy-and-the-logic-of-platforms.

Giroux, Henry A. “Pedagogy against the Dead Zones of the Imagination.” *Transformations: The Journal of Inclusive Scholarship and Pedagogy*, vol. 26, no. 1, 2016, pp. 26-30.

Gold, Matthew K. “The Digital Humanities Moment.” *Debates in the Digital Humanities*, edited by Matthew K. Gold, University of Minnesota Press, 2012, dhdebates.gc.cuny.edu/debates/text/2.

Goldschmidt, Mary. “Teaching Writing in the Disciplines: Student Perspectives on Learning Genre.” *Teaching & Learning Inquiry*, vol. 2, no. 2, 2014, pp. 25-40.

Goldstone, Andrew. “Teaching Quantitative Methods: What Makes It Hard (in Literary Studies).” In *Debates in the Digital Humanities 2019*, edited by Matthew K. Gold and

Lauren F. Klein, University of Minnesota Press, 2019, dhdebates.gc.cuny.edu/read/untitled-f2acf72c-a469-49d8-be35-67f9ac1e3a60/section/620caf9f-08a8-485e-a496-51400296ebcd#ch19.

Golumbia, David. *The Cultural Logic of Computation*. Harvard University Press, 2009.

———. "Death of a Discipline." *differences: a Journal of Feminist Cultural Studies*, vol. 25, no. 1, 2014, pp. 156-176.

Gorzelsky, Gwen, Dana Lynn Driscoll, Joe Paszek, Ed Jones, and Carol Hayes. "Cultivating Constructive Metacognition: A New Taxonomy for Writing Studies." In *Critical Transitions: Writing and the Question of Transfer*, edited by Chris M. Anson and Jessie L. Moore, WAC Clearinghouse, 2016, pp. 215-246.

Govind, Madhu. "What Learning Python Taught Me About Computer Science Education for Young Children." *EdSurge*, 16 June 2020, www.edsurge.com/news/2020-06-16-what-learning-python-taught-me-about-computer-science-education-for-young-children.

Graham, Mark, Matthew Zook, and Andrew Boulton. "Augmented Reality in Urban Places: Contested Content and the Duplicity of Code." *Transactions of the Institute of British Geographers*, vol. 38, no. 3, 2013, pp. 464-479.

Haas, Christina. *Writing Technology: Studies on the Materiality of Literacy*. Lawrence Erlbaum Associates, 1996.

Hall, E. Ashley, Jane Danielewicz, and Jennifer Ware. "Designs for Writing: A Metacognitive Strategy for Iterative Drafting and Revising." In *Using Reflection and Metacognition to Improve Student Learning: Across the Disciplines, Across the Academy*, edited by Matthew Kaplan, Naomi Silver, Danielle Lavaque-Manty, and Deborah Meizlish, Stylus Publishing, 2013, pp. 147-174.

Hallnäs, Lars and Johan Redström. "Slow Technology: Designing for Reflection." *Personal and Ubiquitous Computing*, vol. 5, no. 3, 2001, pp. 201-212.

Hankins, Gabriel. "Correspondence: Theory, Practice, and Horizons." In *Literary Studies in the Digital Age: An Evolving Anthology*, edited by Kenneth Price, Ray Siemens, Dene Grigar, and Elizabeth Lorang. MLA Commons, 2015. dlsanthology.mla.hcommons.org/correspondence-theory-practice-and-horizons/.

Hanstedt, Paul. "Might This Be the Beginning of Education?" *Inside Higher Ed*, 28 April 2020, www.insidehighered.com/views/2020/04/28/how-pandemic-might-create-better-educational-opportunities-opinion.

Hawisher, Gail. “Position Statement.” Computers and Writing Town Hall Meetings 1998. *Kairos: A Journal of Rhetoric, Technology, and Pedagogy*, vol. 3, no. 2, 1998, kairos.technorhetoric.net/3.2/coverweb/statements/hawisher.html.

Hao, Karen. “We read the paper that forced Timnit Gebru out of Google. Here’s what it says.” *MIT Technology Review*, 4 Dec. 2020, www.technologyreview.com/2020/12/04/1013294/ google-ai-ethics-research-paper-forced-out-timnit-gebru/.

Hayden, Wendy. “And Gladly Teach: The Archival Turn’s Pedagogical Turn.” *College English*, vol. 80, no. 2, 2017, pp. 133-158.

Hayles, N. Katherine. *My Mother Was a Computer: Digital Subjects and Literary Texts*. University of Chicago Press, 2005.

———. “Print is Flat, Code is Deep: The Importance of Media-Specific Analysis.” *Poetics Today*, vol. 25, no. 1, 2004, pp. 67-90.

Haynes, Cynthia. “Afterword: <meta> Casuistic Code.” In *From A to <A>: Keywords of Markup*, edited by Bradley Dilger and Jeff Rice, University of Minnesota Press, 2010, pp. 228-235.

Hickman, Leo. “How Algorithms Rule the World.” *The Guardian*, 1 July 2013, www.theguardian.com/science/2013/jul/01/how-algorithms-rule-world-nsa.

Hicks, Mar. “Built to Last.” *Logic*, vol. 11, 31 August 2020, logicmag.io/care/built-to-last/.

Hirsch, Brett. “</Parentheses>: Digital Humanities and the Place of Pedagogy.” In *Digital Humanities Pedagogy: Practices, Principles and Politics*, edited by Brett Hirsch. Open Book Publishers, 2012, pp. 3-30.

Hitchcock, Tim. Keynote to the 2014 British Literary Labs Symposium. The British Library, britishlibrary.typepad.co.uk/files/tim_hitchcock_transcript-1.txt. Accessed 6 Dec. 2020.

Holmes, Steve. “ ‘Can We Name the Tools?’ Ontologies of Code, Speculative Techné and Rhetorical Concealment.” Computational Culture: A Journal of Software Studies, no. 5, 2016, computationalculture.net/can-we-name-the-tools-ontologies-of-code-speculative-techne-and-rhetorical-concealment/.

“Honors Courses.” University of Kentucky Lewis Honors College. www.uky.edu/honors/honors-seminars. Accessed 15 Jan. 2022.

Hsu, Jeremy. “Secret Computer Code Threatens Science.” *Scientific American*, 13 April 2012, www.scientificamerican.com/article/secret-computer-code-threatens-science/. Accessed

26 March 2019.

Huitfeldt, Claus. “Markup Technology and Textual Scholarship.” In *Digital Critical Editions*, edited by Daniel Apollon and Claire Belisle, University of Illinois Press, pp. 157-178.

Hunter, Rik. “The Literacy of Proceduracy: A Conversation with Annette Vee.” *HASTAC*, 18 Sep. 2009, www.hastac.org/blogs/rikhunter/2009/09/18/literacy-proceduracy-conversation-annette-vee.

Ide, Nancy M. and C.M Sperberg-McQueen. “The TEI: History, Goals, Future.” *Computers and the Humanities*, vol. 29, no. 1, 1995, pp. 5-15.

International Data Corporation. *Worldwide Developer Census, 2020: Infrastructure Developers Dominate Part-Time Developer Segment* (abstract). March 2021. www.idc.com/getdoc.jsp?containerId=US47513521.

Iwao, Emma Haruka. “Pi in the sky: Calculating a record-breaking 31.4 trillion digits of Archimedes’ constant on Google Cloud.” *Google Cloud Blog*, 14 March 2019, cloud.google.com/blog/products/compute/calculating-31-4-trillion-digits-of-archimedes-constant-on-google-cloud. Accessed 15 March 2019.

Jackson, Lauren and Desiree Ibekwe. “Jack Dorsey on Twitter’s Mistakes.” *The New York Times*, 7 Aug. 2020, www.nytimes.com/2020/08/07/podcasts/the-daily/Jack-dorsey-twitter- trump.html.

Jockers, Matthew. *Macroanalysis: Digital Methods and Literary History*. University of Illinois Press, 2013.

Jones, Heather. *Violence Against Prisoners of War in the First World War: Britain, France and Germany, 1914-1920*. Cambridge University Press, 2011.

Jones, John and Lavinia Hirsu. Introduction to *Rhetorical Machines: Writing, Code, and Computational Ethics*, edited by John Jones and Lavinia Hirsu, University of Alabama Press, 2019, pp. 1-10.

Kapur, Manu. “Examining Productive Failure, Productive Success, Unproductive Failure, and Unproductive Success in Learning.” *Educational Psychologist*, vol. 51, no. 2, 2016, pp. 289-299.

Kirschenbaum, Matthew. “Digital Humanities As/Is a Tactical Term.” In *Debates in the Digital Humanities*, edited by Matthew K. Gold, University of Minnesota Press, 2012, dhdebates.gc.cuny.edu/read/untitled-88c11800-9446-469b-a3be-3fdb36bfbd1e/section/c0b0a8ee-95f0-4a9c-9451-e8ad168e3db5#ch23.

———. “What Is Digital Humanities and What’s It Doing in English Departments?” In *Debates in the Digital Humanities*, edited by Matthew K. Gold, University of Minnesota Press, 2012, dhdebates.gc.cuny.edu/read/untitled-88c11800-9446-469b-a3be-3fdb36bfbd1e/section/f5640d43-b8eb-4d49-bc4b-eb31a16f3d06#ch01.

———. “Hello Worlds.” *The Chronicle of Higher Education*, 23 Jan. 2009, www.chronicle.com/article/Hello-Worlds/5476.

Kirschner, Paul A. and Pedro De Bruyckere. “The Myths of the Digital Native and the Multitasker.” *Teaching and Teacher Education*, vol. 67, 2017, pp. 135-142.

Klein, Lauren F. “Code.” In *Digital Pedagogy in the Humanities*, edited by Rebecca Frost Davis, Matthew K. Gold, Katherine D. Harris, and Jentery Sayers, Modern Language Association, digitalpedagogy.hcommons.org/keyword/Code.

Ko, Amy J. “Critical CS Education Pedagogy.” In *Critically Conscious Computing: Methods for Secondary Education*, edited by Amy J. Ko, Anne Beitlers, Brett Wortzman, Matt Davidson, Alannah Oleson, Mara Kirdani-Ryan, Stefania Druga, and Jayne Everson, 2022, criticallyconsciouscomputing.org/pedagogy.

———. “Encoding Information.” .” In *Critically Conscious Computing: Methods for Secondary Education*, edited by Amy J. Ko, Anne Beitlers, Brett Wortzman, Matt Davidson, Alannah Oleson, Mara Kirdani-Ryan, Stefania Druga, and Jayne Everson, 2022, criticallyconsciouscomputing.org/information.

Ko, Amy J., et. al. “It Is Time for More Critical CS Education.” *Communications of the ACM*, vol. 63, no. 11, 2020, cacm.acm.org/magazines/2020/11/248199-it-is-time-for-more-critical- cs-education/fulltext.

Koh, Adeline. “A Letter to the Humanities: DH Will Not Save You.” In *Disrupting the Digital Humanities*, edited by Dorothy Kim and Jesse Stommel, Punctum Book, 2018, pp. 39-48.

———. “Niceness, Building, and Opening the Genealogy of the Digital Humanities: Beyond the Social Contract of Humanities Computing.” *differences: a Journal of Feminist Cultural Studies*, vol. 25, no. 1, 2014, pp. 93-106.

Kraus, Kari. “Conjectural Criticism: Computing Past and Future Texts.” *Digital Humanities Quarterly*, vol. 3., no. 4, 2009, www.digitalhumanities.org/dhq/vol/3/4/000069/000069.html.

Kress, Gunther and Theo van Leuuwen. *Reading Images: The Grammar of Visual Design* 2nd edition. Routledge, 1996.

Kurlinkus, William C. *Nostalgic Design: Rhetoric, Memory, and Democratizing Technology*. University of Pittsburgh Press, 2018.

Laiacona, Nick. "FairCopy v.0.11." Vimeo, 18 Aug. 2021, vimeo.com/showcase/7909342/video/588847869. Accessed 18 Dec. 2021.

———. "New Features in FairCopy v0.8.0." Vimeo, 3 Feb. 2021, vimeo.com/showcase/7909342/video/508098694. Accessed 18 Dec. 2021.

Lauer, Janice M. *Invention in Rhetoric and Composition*. Parlor Press, 2004.

Lavin, Matthew. "Why Digital Humanists Should Emphasize Situated Data over Capta." *Digital Humanities Quarterly*, vol. 15, no. 2, 2021, digitalhumanities.org/dhq/vol/15/2/000556/000556.html.

Lecky, Kat. "Humanizing the Interface." In *Disrupting the Digital Humanities*, edited by Dorothy Kim and Jesse Stommel, Punctum Books, 2018, pp. 369-376.

Levin, Sam. "Google accused of 'extreme' gender pay discrimination by US labor department." *The Guardian*, 7 April 2017, www.theguardian.com/technology/2017/apr/07/google-pay-disparities-women-labor-department-lawsuit.

Levine, Caroline. *Forms: Whole, Rhythm, Hierarchy, Network*. Princeton University Press, 2015.

Lindgren, Chris Aaron. "Writing With Data: A Study of Coding on a Data-Journalism Team." *Written Communication* vol. 38, no. 1, 2021, pp. 114-162.

Lockridge, Tim and Derek Van Ittersum. *Writing Workflows: Beyond Word Processing*. University of Michigan Press, 2020. digitalrhetoriccollaborative.org/writing-workflows/index.html.

Long, Seth and James Baker. "The Elusive Digital / Critical Synthesis." In *Debates in the Digital Humanities 2019*, edited by Matthew K. Gold and Lauren F. Klein, University of Minnesota Press, 2019, dhdebates.gc.cuny.edu/read/untitled-f2acf72c-a469-49d8-be35-67f9ac1e3a60/section/72423b53-060d-4927-aa0b-36c90b03152c#ch17.

Lunsford, Andrea. "Our Semi-Literate Youth? Not So Fast." *Stanford Study of Writing*, ssw.stanford.edu/sites/g/files/sbiybj10266/f/OPED_Our_Semi-Literate_Youth.pdf.

Lyubchenko, Irina. "Resisting Clarity/Highlighting Form: Comparing Vanguard Approaches in Poetry and Programming." *Leonardo*, vol. 53, no. 5, 2020, pp. 547-551.

Maher, Jennifer Helene, Helen J. Burgess, and Tim Menzies. "Good Computing with Big Data." In *Rhetorical Machines: Writing, Code, and Computational Ethics*, edited by John Jones

and Lavinia Hirsu, University of Alabama Press, 2019, pp. 190-211.

Manoff, Marlene. "Theories of the Archive Across the Disciplines." *Portal: Libraries and the Academy*, vol. 4, no. 1, 2004, pp. 9-25.

Manovich, Lev. "The Algorithms of Our Lives." *The Chronicle of Higher Education*, 16 Dec. 2013, www.chronicle.com/article/The-Algorithms-of-Our-Lives-/143557/.

Mao, LuMing. "Thinking Beyond Aristotle: The Turn to How in Cultural Rhetoric." *PMLA*, vol. 129, no. 3, 2014, pp. 448-455.

Marcinowski, Peter. "Why is knowing how to code necessary?" *Introduction to Digital Humanities: English 389 at Emory University*. 12 Sep. 2011, www.briancroxall.net/dh/2011/09/12/why-is-knowing-how-to-code-necessary/.

Marino, Mark C. "Critical Code Studies." *Electronic Book Review*, 4 Dec. 2006, electronicbookreview.com/essay/critical-code-studies/.

———. *Critical Code Studies*. The MIT Press, 2020.

Marlow, Jennifer and James P. Purdy. *Are We There Yet? Computers and the Teaching of Writing in American Higher Education—20 Years Later*. Computers and Composition Digital Press and Utah State University Press, 2021. ccdigitalpress.org/book/arewethereyet/.

Mateas, Michael. "Procedural Literacy: Educating the New Media Practitioner." *On the Horizon*, vol. 13, no. 2, 2005, pp. 101-111.

Mateas, Michael and Nick Montfort. "A Box, Darkly: Obfuscation, Weird Languages, and Code Aesthetics." In *Proceedings of the 6th Annual Digital Arts and Culture Conference*, 2005, pp. 144-153

Mayer-Schönberger, Viktor and Kenneth Cukier. *Big Data: A Revolution That Will Transform How We Live, Work, and Think*. John Murray, 2013.

Mazur, Eric. "Peer Instruction for Active Learning." *YouTube*, uploaded by Serious Science, 18 June 2014, www.youtube.com/watch?v=Z9orbxoRofI.

McAfee, Andrew. "Big Data's Biggest Challenge? Convincing People NOT to Trust Their Judgment." *Harvard Business Review*, 9 Dec. 2013, /hbr.org/2013/12/big-datas-biggest-challenge-convincing-people-not-to-trust-their-judgment.

McCarl, Clayton. "Editing the Eartha M.M. White Collection: An Experiment in Engaging Students in Archival Research and Editorial Practice." *The Journal of Academic*

Librarianship, vol. 44, no. 4, 2018, pp. 527-537.

McCarty, Willard. *Humanities Computing*. Palgrave Macmillan, 2005.

———. "Humanities Computing: Essential Problems, Experimental Practice." *Literary and Linguistic Computing*, vol. 17, no. 1, 2002, pp. 103-125.

McClain, Kathryn and Jennifer Murray. "An Analysis of Job Listings on rhetmap.org for Junior Scholars: Academic Years Spanning 2013-2014, 2014-2015, and 2015-2016." *Rhet Map: Mapping Rhetoric and Composition*, Nov. 2016, rhetmap.org/doc/2013to2016rhetmap report.pdf.

McCloud, Susan H. and Eric Miraglia. "Writing Across the Curriculum in a Time of Change." In *WAC for the New Millennium: Strategies for Continuing Writing-Across-the-Curriculum Programs*, edited by Susan H. McLeod, Eric Miraglia, Margot Soven, and Christopher Thaiss. National Council of Teachers of English, 2001, pp. 1-27.

McCulloch, Gretchen. "Coding Is for Everyone—as Long as You Speak English." *Wired*, 8 April 2019, www.wired.com/story/coding-is-for-everyoneas-long-as-you-speak-english/.

McGann, Jerome. *Radiant Textuality: Literature After the World Wide Web*. Palgrave Macmillan, 2001.

McGrail, Anne B. "The 'Whole Game': Digital Humanities at Community Colleges." In *Debates in the Digital Humanities 2016*, edited by Lauren F. Klein and Matthew K. Gold, University of Minnesota Press, 2016, dhdebates.gc.cuny.edu/read/untitled/section/ 39a2e421-1588-455a-8828-3709a728126f.

McPherson, Tara. "Why Are the Digital Humanities So White?" In *Debates in the Digital Humanities*, edited by Matthew K. Gold, University of Minnesota Press, 2012, dhdebates. gc.cuny.edu/read/untitled-88c11800-9446-469b-a3be-3fdb36bfbd1e/section/20df8acd-9ab9-4f35-8a5d-e91aa5f4a0ea.

Meinders, Christine, Jana Thompson, Sarah Ciston, and Catherine Griffiths. "Week 3: Feminist AI (Main Thread)." Critical Code Studies Working Group 2020, 3 Feb. 2020, wg20. criticalcodestudies.com/index.php?p=/discussion/87/week-3-feminist-ai-main-thread.

Metz, Cade. "Lawsuit Takes Aim at the Way A.I. Is Built." *The New York Times*, 23 Nov. 2022, www.nytimes.com/2022/11/23/technology/copilot-microsoft-ai-lawsuit.html. Accessed 1 Dec. 2022.

Miller, Carolyn R. "Should We Name the Tools? Concealing and Revealing the Art of Rhetoric." In *The Public Work of Rhetoric: Citizen-Scholars and Civic Engagement*, edited by John M. Ackerman , and David J. Coogan, University of South Carolina Press, 2010, pp. 19-

38.

"Mission Statement." University of Nebraska-Lincoln Department of English. www.unl.edu/english/mission-statement. Accessed 27 June 2018.

Monea, Bethany. "Screen Reading: A Gallery of (Re)Imagined Interfaces." *Kairos: A Journal of Rhetoric, Technology, and Pedagogy*, vol. 24, no. 2, 2020, kairos.technorhetoric.net/24.2/disputatio/monea/index.html.

Montfort, Nick, Patsy Baudoin, John Bell, Ian Bogost, Jeremy Douglass, Mark C. Marino, Michael Mateas, Casey Reas, Mark Sample, and Noah Vawter. *10 PRINT CHR$(205.5+RND(1)); : GOTO 10.* The MIT Press, 2012.

Morris, Sean Michael. "Critical Instructional Design." In *An Urgency of Teachers: The Work of Critical Digital Pedagogy*, edited by Sean Michael Morris and Jesse Stommel. Hybrid Pedagogy, 2018, criticaldigitalpedagogy.pressbooks.com/chapter/critical-pedagogy-and-learning-online/.

Munroe, Randall. "Code Quality." XKCD. 17 April 2015. xkcd.com/1513/.

———. "Code Quality 2." XKCD. 17 June 2016. xkcd.com/1695/.

Nasser, Ramsey. "قلب." Personal Website, nas.sr/قلب. Accessed Sep. 27, 2020.

———. "A Personal Computer for Children of All Cultures." In *Decolonising the Digital: Technology As Cultural Practice*, edited by Josh Harle, Angie Abdilla, and Andrew Newman, Tactical Space Lab, 2018, pp. 21-36, ojs.decolonising.digital/index.php/decolonising_digital/article/view/PersonalComputer/3.

National Research Council. *How People Learn: Brain, Mind, Experience, and School: Expanded Edition*. National Academies Press, 2000.

Niester-Mika, Jennifer. *Critical Experiential Learning And Rhetorical Interventions In New Media Ecologies*. 2014. Wayne State University, PhD dissertation.

Noble, Safiya Umoja. *Algorithms of Oppression: How Search Engines Reinforce Racism*. NYU Press, 2018.

Nyhan, Julianne and Andrew Flinn. *Computation and the Humanities: Towards an Oral History of Digital Humanities.* Springer, 2016.

Obama, Barack. "Computer Science Education Week 2013." *YouTube*, uploaded by The Obama White House, 8 December 2013, youtu.be/yE6IfCrqg3s. Accessed 26 March 2019.

Ong, Water J. *Orality and Literacy: 30th Anniversary Edition*. Routledge, 2012.

ParkeHarrison, Robert and Shana ParkeHarrison. *Summer Arm*. www.parkeharrison.com/bodies-of-work/counterpoint-gray-dawn/284.

Pea, Roy D. and D. Midian Kurland. "On the Cognitive Effects of Learning Computer Programming." *New Ideas in Psychology*, vol. 2, no. 2, 1984, pp. 137-168.

Pettiway, Keon. "Scholar Introduction: Stephen Ramsay is wrong." *HASTAC*, 4 Nov. 2013, www.hastac.org/blogs/kmpettiw/2013/11/04/scholar-introduction-stephen-ramsay-wrong-0.

Pierazzo, Elena. "A Rationale of Digital Documentary Editions." *Literary and Linguistic Computing*, vol. 26, no. 4, 2011, pp. 463-477.

Pinker, Steven. *The Sense of Style: The Thinking Person's Guide to Writing in the 21st Century*. Penguin, 2014.

Pitchler, Alois and Tone Merete Bruvik. "Digital Critical Editing: Separating Encoding from Presentation." In *Digital Critical Editions*, edited by Daniel Apollon, Claire Belisle, and Philippe Regnier. University of Illinois Press, 2014, pp. 179-199.

Plato. *Ion*, translated by Paul Woodruff. In *Plato on Rhetoric and Language: Four Dialogues*, edited by Jean Nienkamp, Hermagoras Press, 1999, pp. 23-35.

Posner, Miriam. "Some things to think about before you exhort everyone to code." *Miriam Posner's Blog*, 29 Feb. 2012, miriamposner.com/blog/some-things-to-think-about-before-you-exhort-everyone-to-code.

———. "What's Next: The Radical, Unrealized Potential of Digital Humanities." In *Debates in the Digital Humanities 2016*, edited by Lauren F. Klein and Matthew K. Gold, University of Minnesota Press, 2016, dhdebates.gc.cuny.edu/read/untitled/section/a22aca14-0eb0-4cc6-a622-6fee9428a357#ch03.

Prensky, Mark. "Digital Natives, Digital Immigrants, Part 1." *On the Horizon*, vol. 9, no. 5, 2001, pp. 1-6.

Prescott, Andrew. "Beyond the Digital Humanities Center: The Administrative Landscapes of the Digital Humanities." In *A New Companion to Digital Humanities*, edited by Susan Schreibman, Ray Siemens, and John Unsworth. Blackwell, 2016, pp. 461-475.

Price, Colin B. and Ruth M. Price-Mohr. "Stories Children Write While Coding: A Cross-Disciplinary Approach for the Primary Classroom." *Cambridge Journal of Education*,

vol. 48, no. 6, pp. 735-747.

Ramsay, Stephen. "The Hermeneutics of Screwing Around; or What You Do with a Million Books." In *Pastplay: Teaching and Learning History with Technology*, edited by Kevin Kee, University of Michigan Press, 2014, quod.lib.umich.edu/d/dh/12544152.0001.001/1:5/--pastplay-teaching-and-learning-history-with-technology?g=dculture;rgn=div1;view=fulltext;xc=1#5.1.

———. "On Building." *Stephen Ramsay*, 2011, web.archive.org/web/20160716135432 /http://stephenramsay.us:80/text/2011/01/11/on-building. Accessed 10 Mar. 2018.

———. Response to "Why is knowing how to code necessary?" *Introduction to Digital Humanities: English 389 at Emory University*. 13 Sep. 2011, www.briancroxall.net/dh/2011/09/12/why-is-knowing-how-to-code-necessary/#comment-180.

———. "Who's In and Who's Out." *Stephen Ramsay*, 2011, web.archive.org/web/20160906122605/http://stephenramsay.us:80/text/2011/01/08/whos-in-and-whos-out. Accessed 10 March 2018.

Ramsay, Stephen and Geoffrey Rockwell. "Developing Things: Notes toward an Epistemology of Building in the Digital Humanities." In *Debates in the Digital Humanities*, edited by Matthew K. Gold, University of Minnesota Press, 2012, dhdebates.gc.cuny.edu/debates/text/11.

Raley, Rita. "Interferences: [Net.Writing] and the Practice of Codework." *Electronic Book Review*, 8 Sep. 2002, electronicbookreview.com/essay/interferences-net-writing-and-the-practice-of-codework/.

Reider, David M. "Programming Is the New Ground of Writing." *Enculturation: A Journal of Rhetoric, Writing, and Culture*, 10 Oct. 2012, enculturation.net/node/5267.

Renear, Allen H. "Text Encoding." In *A Companion to Digital Humanities*, edited by Susan Schreibman, Ray Siemens, and John Unsworth, Blackwell, 2004, digitalhumanities.org:3030/companion/view?docId=blackwell/9781405103213/9781405103213.xml&chunk.id=ss1-3-5&toc.depth=1&toc.id=ss1-3-5&brand=9781405103213_brand.

Resnick, Mitchel. "Learn to Code, Code to Learn." *EdSurge*, 8 May 2013, www.edsurge.com/news/2013-05-08-learn-to-code-code-to-learn.

Rice, Jenny and Jeff Rice. "Pop-Up Archives." In *Rhetoric and the Digital Humanities*, edited by Jim Ridolfo and William Hart-Davidson, University of Chicago Press, 2015, pp. 245-254.

Ridolfo, Jim and Dànielle Nicole DeVoss. "Composing for Recomposition: Rhetorical Velocity and Delivery." *Kairos: A Journal of Rhetoric, Technology, and Pedagogy*, vol. 13, no. 2,

2009, kairos.technorhetoric.net/13.2/topoi/ridolfo_devoss/index.html.

Ridolfo, Jim and William Hart-Davidson. Introduction to *Rhetoric and the Digital Humanities*, edited by Jim Ridolfo and William Hart-Davidson, University of Chicago Press, 2015, pp. 1-12.

Risam, Roopika. *New Digital Worlds: Postcolonial Digital Humanities in Theory, Praxis, and Pedagogy*. Northwestern University Press, 2019.

———. "Where Have All the DH Jobs Gone?" Personal Website, 15 Sep. 2013, roopika risam.com/uncategorized/where-have-all-the-dh-jobs-gone/. Accessed 24 Mar. 2018.

Robertson, Campbell. "They Were Promised Coding Jobs in Appalachia. Now They Say It Was a Fraud." *The New York Times*, 12 May 2019, www.nytimes.com/2019/05/12/us/mined-minds-west-virginia-coding.html.

Rogers, Jeff. "Lines in the Sand or Growing Pains: Stephen Ramsay and the Digital Humanities at the 2011 MLA." *Townsend Center for the Humanities*. University of California, Berkeley, 13 Jan. 2011, townsendcenter.berkeley.edu/blog/lines-sand-or-growing-pains -stephen-ramsay-and-digital-humanities-2011-mla.

Rushkoff, Douglas. *Program or Be Programmed: Ten Commands for a Digital Age*. OR Books, 2010.

Sageev, Pneena and Carol J. Romanowski. "A Message from Recent Engineering Graduates in the Workplace: Results of a Survey on Technical Communication Skills." *Journal of Engineering Education*, vol. 90, no. 4, 2001, pp. 685-693.

Sample, Mark. "Criminal Code: Procedural Logic and Rhetorical Excess in Videogames." *Digital Humanities Quarterly*, vol. 7, no. 1, 2013, www.digitalhumanities.org/dhq/vol/7/ 1/000153/000153.html.

———."Digital Humanities Sessions at the 2011 MLA." *samplereality*, 9 Nov. 2010, www. samplereality.com/2010/11/09/digital-humanities-sessions-at-the-2011-mla/.

———. "The Poetics of Non-Consumptive Reading." *samplereality*, 22 May 2013, www. samplereality.com/2013/05/22/the-poetics-of-non-consumptive-reading/.

Sayers, Jentry. "Introduction: 'I Don't Know All the Circuitry.' " In *Making Things and Drawing Boundaries: Experiments in the Digital Humanities*, edited by Jentery Sayers, University of Minnesota Press, 2017, dhdebates.gc.cuny.edu/read/untitled-aa1769f2-6c55-485a-81af -ea82cce86966/section/7d8fca82-c6ca-480f-bf17-1df4a2cdb577#intro.

Schlesinger, Ari. “A Feminist & A Programmer.” HASTAC Blog, 13 Dec. 2013, www.hastac.org /blogs/ari-schlesinger/2013/12/13/feminist-programmer.

———. “A Feminist Programming Language?” FemTechNet Blog, 14 July 2014, femtechnet.org/2014/07/a-feminist-programming-language/.

Schlesinger, Ari, Kenton P. O’Hara, and Alex S. Taylor. “Let’s Talk About Race: Identity, Chatbots, and AI.” *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, no. 315, 2018, https://dl.acm.org/doi/10.1145/3173574.3173889.

Schmidt, Benjamin. “Do Digital Humanists Need to Understand Algorithms?” In *Debates in the Digital Humanities 2016*, edited by Lauren F. Klein and Matthew K. Gold, University of Minnesota Press, 2016, dhdebates.gc.cuny.edu/read/untitled/section/557c453b-4abb-48ce-8c38-a77e24d3f0bd#ch48.

Schneider, Bertrand, Paulo Blikstein, and Roy Pea. “The Flipped, Flipped Classroom.” *The Stanford Daily*, 5 Aug. 2013, www.stanforddaily.com/2013/08/05/the-flipped-flipped-classroom/. Accessed 22 Jan. 2022.

Schreibman, Susan, Ray Siemens, and John Unsworth. “The Digital Humanities and Humanities Computing: An Introduction.” In *A Companion to Digital Humanities*, edited by Susan Schreibman, Ray Siemes, and John Unsworth, Blackwell, 2004, www.digitalhumanities.org/companion/.

———. Preface. In *A New Companion to Digital Humanities*, edited by Susan Schreibman, Ray Siemens, and John Unsworth, Blackwell, 2016, pp. Xvii-xviii.

Schwartz, Daniel L. and John D. Bransford. “A Time for Telling.” *Cognition and Instruction*, vol. 16, no. 4, 1998, pp. 475-522.

Selber, Stuart. *Multiliteracies for a Digital Age*. Southern Illinois University Press, 2004.

Selfe, Cynthia. “The Movement of Air, the Breath of Meaning: Aurality and Multimodal Composing.” *College Composition and Communication*, vol. 60, no. 4, 2009, pp. 616-663.

Selfe, Cynthia and Richard Selfe. “The Politics of the Interface: Power and its Exercise in Electronic Contact Zones.” *College Composition and Communication*, vol. 45, no. 4, 1994, pp. 480-504.

Seraphin, Sally B., et al. “A Conceptual Framework for Non-Disposable Assignments: Inspiring Implementation, Innovation, and Research.” *Psychology Learning and Teaching*, vol. 18, no. 1, 2019, pp. 84-97.

Sheffield, Jenna Pack. “Thinking Beyond Tools: Writing Program Administration and Digital Literacies.” *Computers and Composition Online*, 2015, cconlinejournal.org/sheffield/Index.html.

Shipka, Jody. *Toward a Composition Made Whole*. University of Pittsburgh Press, 2011.

Silver, Naomi. “Reflective Pedagogies and the Metacognitive Turn in College Teaching.” In *Using Reflection and Metacognition to Improve Student Learning: Across the Disciplines, Across the Academy*, edited by Matthew Kaplan, Naomi Silver, Danielle Lavaque-Manty, and Deborah Meizlish, Stylus Publishing, 2013, pp. 1-17.

Smiley, Lauren. “Can You Teach a Coal Miner to Code?” *Wired*, 18 Nov. 2015, www.wired.com/2015/11/can-you-teach-a-coal-miner-to-code/.

Smith, Kevin G. “Genre: Review, Student ID: to.fr.” *Markup in the Writing Classroom*. markup.kevingeraldsmith.com/php/project_page.php?id=to.fr&proj=2&genre=review.

———. *(Re)making/(Re)marking: Genre and Markup in the Writing Classroom*. 2018. Northeastern University, PhD dissertation.

Smith, Martha Nell. “The Human Touch Software of the Highest Order: Revisiting Editing as Interpretation.” *Textual Cultures: Texts, Contexts, Interpretation*, vol. 2, no. 1, pp. 1-15.

Sontag, Susan. *Against Interpretation and Other Essays*. Picador, 1966.

Sperberg-McQueen, C. M. 2019. “Playing for Keeps: The role of modeling in the humanities.” In *The Shape of Data in Digital Humanities: Modeling Texts and Text-Based Resources*, edited by Julia Flanders and Fortis Jannidis, 283–310. New York: Routledge.

Spring, Michael B. “The Origin and Use of Copymarks in Electronic Publishing.” *Journal of Documentation*, vol. 45, no. 2, 1989, pp. 110-123.

Stadler, Peter, Marcel Illetschko, and Sabine Seifert. “Towards a Model for Encoding Correspondence in the TEI: Developing and Implementing `<corresp>`.” *Journal of the Text Encoding Initiative*, no. 9, Sep. 2016-Dec. 2017, journals.openedition.org/jtei/1433.

Stagliano, Anthony. “Metis in Code: CV Dazzle and the Wily Encounter with Code Libraries.” In *Rhetorical Machines: Writing, Code, and Computational Ethics*, edited by John Jones and Lavinai Hirsu, University of Alabama Press, 2019, pp. 170-189.

Steiner, Christopher. *Automate This: How Algorithms Came to Rule Our World*. Portfolio, 2012.

Stiegler, Bernard. “Die Aufklärung in the Age of Philosophical Engineering.” *Computational Culture*, no. 2, 2012, computationalculture.net/die-aufklarung-in-the-age-of-philosophical-engineering/.

Stommel, Jesse. “Critical Digital Pedagogy: A Definition.” *Hybrid Pedagogy*, 18 Nov. 2014, hybridpedagogy.org/critical-digital-pedagogy-definition/.

———. “The Public Digital Humanities.” In *Disrupting the Digital Humanities*, edited by Dorothy Kim and Jesse Stommel, Punctum Books, 2018, pp. 79-90.

Svensson, Patrik. “Humanities Computing as Digital Humanities.” *Digital Humanities Quarterly*, vol. 3, no. 3, 2009, digitalhumanities.org/dhq/vol/3/3/000065/000065.html.

Takayoshi, Pamela. “The Shape of Electronic Writing: Evaluating and Assessing Computer-Assisted Writing Processes and Products.” *Computers and Composition*, vol. 13, no. 2, 1996, pp. 245-257.

Tanaka, Kurtis, et al. *Teaching with Primary Sources: Looking at the Support Needs of Instructors*. Ithaka S+R, 2021. doi: 10.18665/sr.314912.

Tenen, Dennis. “Blunt Instrumentalism: On Tools and Methods.” *Debates in the Digital Humanities*, edited by Matthew K. Gold and Lauren F. Klein, University of Minnesota Press, 2016, dhdebates.gc.cuny.edu/debates/text/60.

Text Encoding Initiative. “16 Linking, Segmentation, and Alignment.” *TEI P5 Guidelines*, 19 Aug. 2020, tei-c.org/release/doc/tei-p5-doc/en/html/SA.html.

Theimer, Kate. “Archives in Context and as Context.” *Journal of Digital Humanities*, vol. 1, no. 2, 2012, journalofdigitalhumanities.org/1-2/archives-in-context-and-as-context-by-kate-theimer/.

Thomas, William G. “What We Think We Will Build and What We Build in Digital Humanities.” *Journal of Digital Humanities*, vol. 1, no. 1, Winter 2011, journalofdigitalhumanities.org/1-1/what-we-think-we-will-build-and-what-we-build-in-digital-humanities-by-will-thomas/.

Thompson, Clive. “The Next Big Blue-Collar Job is Coding.” *Wired*, 08 Feb. 2017, www.wired.com/2017/02/programming-is-the-new-blue-collar-job/.

Tinnell, John. “Grammatization: Bernard Stiegler’s Theory of Writing and Technology.” *Computers and Composition*, vol. 37, 2015, pp. 132-146.

———. “Post-Media Occupations for Writing Theory: From Augmentation to Autopoeisis.” In *Ecology, Writing Theory, and New Media: Writing Ecology*, edited by Sidney I. Dobrin,

Routledge, 2012, pp. 122-142.

Trimble, Thomas and Adrienne Jankens. "Using Taxonomies of Metacognitive Behaviors to Analyze Student Reflection and Improve Teaching Practice." *Pedagogy: Critical Approaches to Teaching Literature, Language, Composition, and Culture*, vol. 19, no. 3, 2019, pp. 433-454.

Underwood, Ted. "A Broader Purpose." *The Stone and the Shell: Using Large Digital Libraries to Advance Literary History* (personal website), 4 Jan. 2018, tedunderwood.com/2018/01/04/a-broader-purpose/.

Uskoković, Vuc. "Flipping the Flipped: The Co-Creational Classroom." *Research and Practice in Technology Enhanced Learning*, vol. 13, 2018, telrp.springeropen.com/articles/10.1186 /s41039-018-0077-9.

VanKooten, Crystal. "Identifying Components of Meta-Awareness about Composition: Toward a Theory and Methodology for Writing Studies." *Composition Forum*, vol. 33, 2016, compositionforum.com/issue/33/meta-awareness.php.

Vee, Annette. *Coding Literacy: How Computer Programming is Changing Writing*. MIT Press, 2017.

———. "Full Stack Rhetoric." In *Rhetorical Machines: Writing, Code, and Computational Ethics*, University of Alabama Press, 2019, pp. 237-244.

Vee, Anette and James J. Brown, Jr. "Rhetoric Special Issue Introduction." *Computational Culture: A Journal of Software Studies*, no. 5, 2016, computationalculture.net/rhetoric-special-issue-editorial-introduction/.

Villarroela, Verónica, Susan Bloxham, Daniela Bruna, Carola Bruna, and Constanza Herrera-Seda. "Authentic Assessment: Creating a Blueprint for Course Design." *Assessment and Evaluation in Higher Education*, vol. 43, no. 5, 2018, pp. 840-854.

Vong, Silvia. "A Constructivist Approach for Introducing Undergraduate Students to Special Collections and Archival Research." *RBM: A Journal of Rare Books, Manuscripts, and Cultural Heritage,* vol. 17, no. 2, 2016, pp. 148–171.

Warner, John. *Why They Can't Write: Killing the Five-Paragraph Essay and Other Necessities*. Johns Hopkins University Press, 2018.

Wharton, Robin. "Bend Until It Breaks." In *Disrupting the Digital Humanities*, edited by Dorothy Kim and Jesse Stommel, Punctum Books, 2018, pp. 377-399.

———. “Building in the Humanities Isn’t New.” In *Critical Digital Pedagogy: A Collection*, edited by Jesse Stommel, Chris Friend, and Sean Michael Morris, Hybrid Pedagogy, 2020, cdpcollection.pressbooks.com/chapter/building-in-the-humanities-isnt-new/.

Wiggins, Grant. 1990. “The Case for Authentic Assessment.” *Practical Assessment, Research, and Evaluation*, vol. 2, no. 2 1990, scholarworks.umass.edu/pare/vol2/iss1/2/.

Wilson, Greg. Preface. *Beautiful Code: Leading Programmers Explain How They Think*. Edited by Andy Oram and Greg Wilson, O'Reilly Media, 2007, pp. Xv-xvi.

Wilson, Noah. “What Ethos Can Tell Us About Conspiracy Theories.” *Gayle Morris Sweetland Digital Rhetoric Collaborative*, 27 July 2020, www.digitalrhetoriccollaborative.org/2020/07/27/what-ethos-can-tell-us-about-conspiracy-theories/.

Wing, Jeannette M. “Computational Thinking.” *Communications of the ACM*, vol. 49, no. 3, 2006, pp. 33-35.

Yancey, Kathleen Blake, Liane Robertson, and Kara Taczak. *Writing Across Contexts: Transfer, Composition, and Sites of Writing*. University Press of Colorado and Utah State University Press, 2014.

Yin, Robert K. *Case Study Research and Applications: Design and Methods* 6th edition. Sage Publications, 2018.

Zimmerman, Barry J. and Adam R. Moylan. “Self-Regulation: Where Metacognition and Motivation Intersect.” In *Handbook of Metacognition in Education*, edited by Douglas J. Hacker, John Dunlosky, and Arthur C. Graesser, Routledge, 2009, pp. 299-315.

Zuboff, Shoshana. *The Age of Surveillance Capitalism: The Fight for a Human Future at the New Frontier of Power*. PublicAffairs, 2019.

Appendix A: Example Primary Source Analysis from Markup-Based FYW Class

I. Markup Scheme

The following elements, attributes, and attribute values were required of students in the Primary Source Analysis. This list was included as part of the syllabus and course requirements in the HTML-based course website that was updated throughout the semester with students' work.

```
<div type="introduction">introduction</div>

<div type="conclusion">conclusion</div>

<p>paragraph</p>

<title level="a">Title of an article, song, image, etc.</title>

<title level="m">Title of a book, film, album, etc.</title>

<title level="j">Title of a newspaper, journal, magazine, or other periodical</title>

<hi rend="italics">italicized text</hi>

<figure rend="link">
    <graphic url="url_here"/>
    <!-- Link directly to primary source after first naming it. Note that this is an "empty"
    element that does not contain any parsed character data, i.e., text. -->
</figure>

<seg type="thesis">
    Tag the overall thesis statement; it can be more than one sentence and more than one part.
    This tag will contain the segment tags for thesis parts.
    <seg type="thesis_part" n="#">
        Based on our reading and discussion about the work and types of thesis statements, tag
        each part of your thesis with an incremental N-value, e.g., 1, 2, 3.
    </seg>
    <!-- Include as few or as many "theis_part" segments as needed. Your thesis should have at
    least two parts. -->
</seg>

<seg type="ev_interp">
    <seg type="evidence">evidence (i.e., observable details) from the primary source</seg>
    <seg type="interpretation">interpretation based on the corresponding evidence</seg>
</seg>
```

```
<!-- At least one time (though I strongly encourage you to do more), match evidence from the primary source with the interpretations you draw from it. -->

<seg type="research_question" xml:id="surname_rq_#">research question</seg>
<!-- Tag all research questions with unique @xml:id identifiers. We will link back to these in later assignments, e.g., in the annotated bibliography. -->

<seg type="complication">text here</seg>
<!-- Tag at least one moment when you complicate the seemingly obvious. →

<seg type="pattern">text here</seg>
<!-- Tag at least one moment when you discuss an exact repetition in the primary source. →

<seg type="strand">text here</seg>
<!-- Tag at least one moment when you discuss a strand, i.e., an inexact or loose repetition, in the primary source. -->

<seg type="binary">
    <seg type="binary_a">your observation or discussion of one-half of a binary</seg>
    <seg type="binary_b">your observation or discussion of the other half of a binary</seg>
</seg>
<!-- Tag your discussion or explanation of at least one binary, i.e., an "organizing contrast," in the primary source. -->

<seg type="anomaly">
    <seg type="anomaly_ps">your observation of an anomaly in the primary source</seg>
    <seg type="anomaly_sig">your explanation or speculation as to the significance</seg>
</seg>
<!-- Tag at least one moment when you discuss an anomaly in the primary source as well as your explanation of the significance of that anomaly.-->
```

II. Example XML-Encoded Text of a Primary Source Analysis

The following example appeared in the syllabus of the first-year writing class. Students' analyses would appear within their corresponding `<TEI>` element later in the corpus file.

```
<div type="example" n="1" xml:id="psa_example_1">
    <head type="student_name" n="1">J. Q. Student</head>
    <head type="instructor_name" n="2">Trey Conatser</head>
    <head type="class" n="3">English 1110.01</head>
    <head type="date" n="4">04 February 2012</head>
    <head type="paper_title" n="5">Primary Source Analysis</head>
    <!-- The <div> and <head> structure above was built out for each student in their <TEI> elements beforehand. They filled in the elements with their information. -->
    <div type="introduction">
        <p>
```

```
        Robert and Shana ParkeHarrison’s <title level="a">Summer Arm</title>
        <figure rend="link">
            <graphic url="https://www.parkeharrison.com/bodies-of-work/counterpoint-gray-
            dawn/284/3"/>
        </figure>
        , a mixed media (though mostly photographic) piece in their <title level="m">
        Counterpoint</title> series, presents us with a mechanical apparatus holding the
        outstretched arm of a man whose body for the most part lies beyond the frame.
        <seg type="pattern">
            Atop the arm grow three clusters of plants, including Black-eyed Susans, a Tiger
            Lily, and small fern stalks.
        </seg>
        <seg type="pattern">
            Four butterflies of varying colors and an insect of an uncertain type fly around
            the flowers in front of the flat, off-white background.
        </seg>
        <seg type="thesis">
            <seg type="thesis_part">
                <title level="a">Summer Arm</title> therefore consists both compositionally
                and conceptually of three major parts: the mechanical, the human, and the
                natural, a triad that the ParkeHarrisons have made the focus of most, if not
                all of their work.
            </seg>
            <seg type="thesis_part">
                In addition to raising questions about the nature of each of these parts,
                <title level="a">Summer Arm</title> leads us to reconsider how much they
                feed into or push back against each other.
            </seg>
        </seg>
    </p>
</div>
<p>
    The apparatus occupies roughly the bottom fourth of the image.
    <seg type="ev_interp">
        <seg type="evidence">
            Hard right angles and sickle-like curves convey a harshness and coldness matched
            by the silver and black of the skeletal pieces. Though obviously mechanical, the
            device also appears to have been assembled idiosyncratically;
            <seg type="pattern">
                the irregular knobs, connectors, and sections
                <!-- This shows how the markup can contain other elements of the scheme, in
                this case an observation about a pattern nested within the already-nested
                structure for connecting evidence with interpretation. -->
            </seg>
            may very well have been scrounged from a scrap pile.
        </seg>
        <seg type="interpretation">
            At best, therefore, the apparatus suggests a moral ambivalence; while it
            unavoidably points to the impersonal, mass fabrication of modern industry, it
```

```
            also represents a creative recycling of available resources, the castoff
            detritus from the engines of consumption and waste.
        </seg>
    </seg>
    <seg type="ev_interp">
        <seg type="interpretation">
            Moreover, the device both threatens and supports the man’s arm.
        </seg>
        <!-- Because of how XPath navigates XML nodes for XSL transformations, the order of
        sibling elements does not affect the program’s ability to find them. This technical
        affordance allows the compositional insight that the interpretation may precede or
        proceed from the evidence, depending on the needs of the moment. -->
        <seg type="evidence">
            Rods taper to needle-like points uncomfortably close to the man’s flesh, and a
            circular component just above the elbow seems to function as a clamp holding the
            arm in place: a buttress, or, conversely, a restraint.
        </seg>
    </seg>
</p>
<p>
    Thus, we question whether the man himself constructed the apparatus or if he simply was
    placed in it. His upturned arm appears hyperextended, and his shirtsleeve has been
    rolled back; overall, the body’s position recalls the act of giving blood or having a
    blood sample taken, in both cases the loss of vivifying, essential fluid. Furthermore,
    the man’s head lends an additional appearance of exhaustion.
    <seg type="anomaly">
        <seg type="anomaly_ps">
            Though we see only the very top of his head, it clearly tilts deeply in the
            direction of the extended arm,
        </seg>
        <seg type="anomaly_sig">
            perhaps compensating for the awkward hyperextension, or perhaps resting out of
            sheer exhaustion. Indeed, we don’t know how long the man has been in the
            apparatus; if the plants actually have grown on his arm, he may very well have
            been in this position for quite a while. This character also appears with arm-
            sprouted plants at other stages of the growth cycle in <title level="a"> Winter
            Arm</title>
            <figure rend="link">
                <graphic url="https://www.parkeharrison.com/bodies-of-work/counterpoint-
                gray-dawn/285/0"/>
            </figure>
            and honeycombs tied to his arm in <title level="a">Spring Arm</title>
            <figure rend="link">
                <graphic url="https://www.parkeharrison.com/bodies-of-work/counterpoint-
                gray-dawn/283/4"/>
            </figure>
            in the <title level="m">Counterpoint Gray Dawn</title> series, though he is not
            bound by an apparatus in either images. If there is an implied story, it is
            mythic and unclear, the man appearing as an archetype across the disjointed
            mises en scène.
```

```
        </seg>
    </seg>
</p>
<p>
    Returning to the question of purpose, the <hi rend="italics">why</hi> of the image's
    representation, we look to the plants themselves.
    <seg type="binary">
        <seg type="binary_a">Utility</seg> does not underwrite their cultivation; rather,
        the <seg type="binary_b">whimsy</seg>
    </seg>
    of their variety indicates that they function more as a sign of the color and bounty of
    summer's flora. Put simply, the man doesn't seem to be accomplishing anything useful in
    strictly instrumental terms. Growing the plants, then, evinces less of
    <seg type="binary">
        <seg type="binary_a">a material goal</seg>&#x2014;human use or natural
        restoration&#x2014;and more of <seg type="binary_b">a ceremonial devotion</seg>.
        <!-- &#x2014; is the unicode hex character code for the em dash. -->
    </seg>
    Because we have no context for the ritual, because instead of using lush scenery the
    image foregrounds the situation atop a depthless, ascetic, white matte, the tone
    strikes the viewer not as celebratory but as elegiac.
</p>
<p>
    Again, the ParkeHarrisons imply narrative through the questions that we're led to ask:
    what is the purpose of the ritual, if it is a ritual in the first place? what does the
    ritual elegize, and if it does elegize something, what led to that loss? At this point
    a narrative of ecological decline or even disaster isn't beyond the pale, and we
    wouldn't be out of line in postulating for the man the role of a minister of a lost
    faith or forgotten religion.
    <seg type="complication">
        Despite the clear differences between the mechanical, the human, and the natural,
        complicating similarities slyly lie behind the organizing contrasts.
        <seg type="strand">
            The vertical lines of the mechanical apparatus continue in the more organic form
            of the plants' stems and leaves beyond the perpendicular horizon of the man's
            arm.
        </seg>
        Compositionally speaking, then, the human either divides the natural from the
        mechanical, or it represents a blend of the two. We often invoke
        <seg type="binary">
            <seg type="binary_a">technology and industry</seg> as antithetical to <seg
            type="binary_b">nature,</seg>
        </seg>
        but in <title level="a">Summer Arm</title> their purpose is specifically to support
        natural growth, which then attracts the additional life of the butterflies and bee-
        like insect. Of course, this use of mechanical technology results not from large-
        scale efforts but from an individual's ritualistic bricolage: a repurposing or even
        subversion of technology's driving ethos.
    </seg>
</p>
```

```
    <div type="conclusion">
        <p>
            <title level="a">Summer Arm</title> leads us to several larger questions about the
            ParkeHarrisons' work.
            <seg type="research_question" xml:id="student_rq_1">
                Despite the implication that industry and technology destroy the natural world,
                to what degree do the ParkeHarrisons present (and endorse) the possibility that
                they can be harnessed as ecological adjuncts?
            </seg>
            <seg type="research_question" xml:id="student_rq_2">
                Like the man in <title level="a">Summer Arm</title>, do the ParkeHarrisons make
                their artworks as elegies or as the kind of patched-together scaffolds on which
                the human may foster growth?
            </seg>
            <seg type="research_question" xml:id="student_rq_3">
                Just what sort of sacrifice or support does nature require from us?
            </seg>
            <seg type="research_question" xml:id="student_rq_4">
                Ultimately, how can we negotiate Summer Arm's various selves&#x2014;aesthetic
                object, abstract symbolism, programmatic allegory, call to action, and, finally,
                material object whose very composition involves the mechanical-human-natural
                triad that it represents?
            </seg>
        </p>
    </div>
</div>
```

III. HTML Views of Transformed XML-Encoded Primary Source Analysis

Students would be able to view their work in two ways on the course website: the unaltered text as it would appear in any word processor or on any website, and a filtered version of the text that drew students' attention to important moves in their writing. The XSL transformations produced both of these views and depended on students using the markup scheme exactly as specified; the XPath navigation would not be able to locate the target elements otherwise. Figures 23 and 24 below respectively show the HTML views for the instructor-generated example primary source analysis (standard and filtered views) and the main menu of the course website for exploring the primary source analysis across all students' work for the assignment.

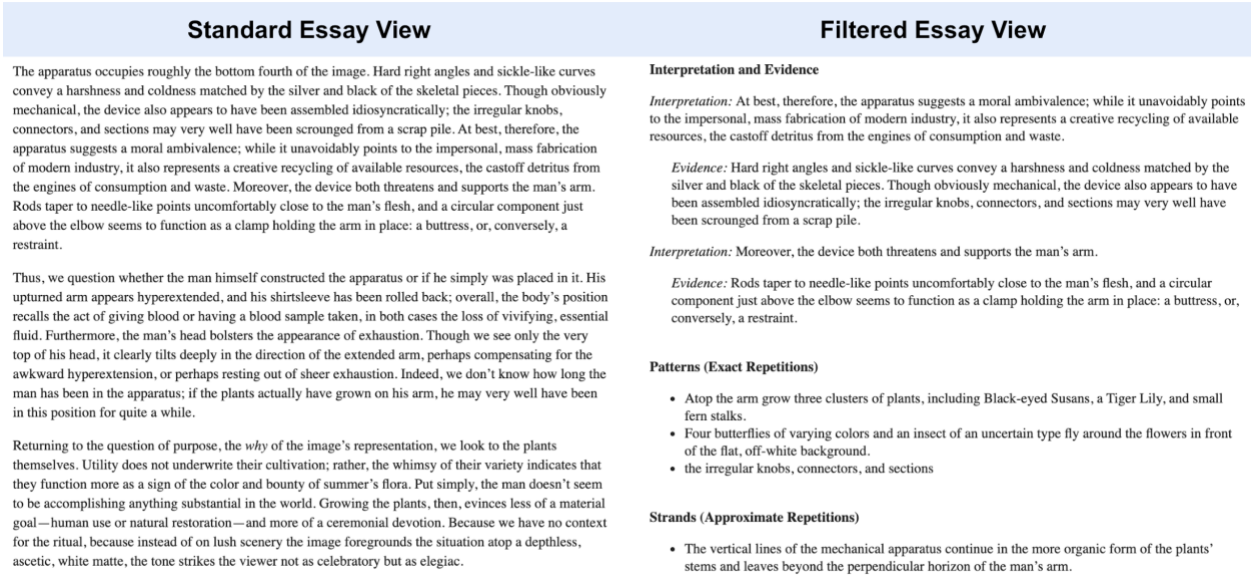


Figure 23: Side-by-side screenshots of standard and filtered essay views on the HTML-based course website

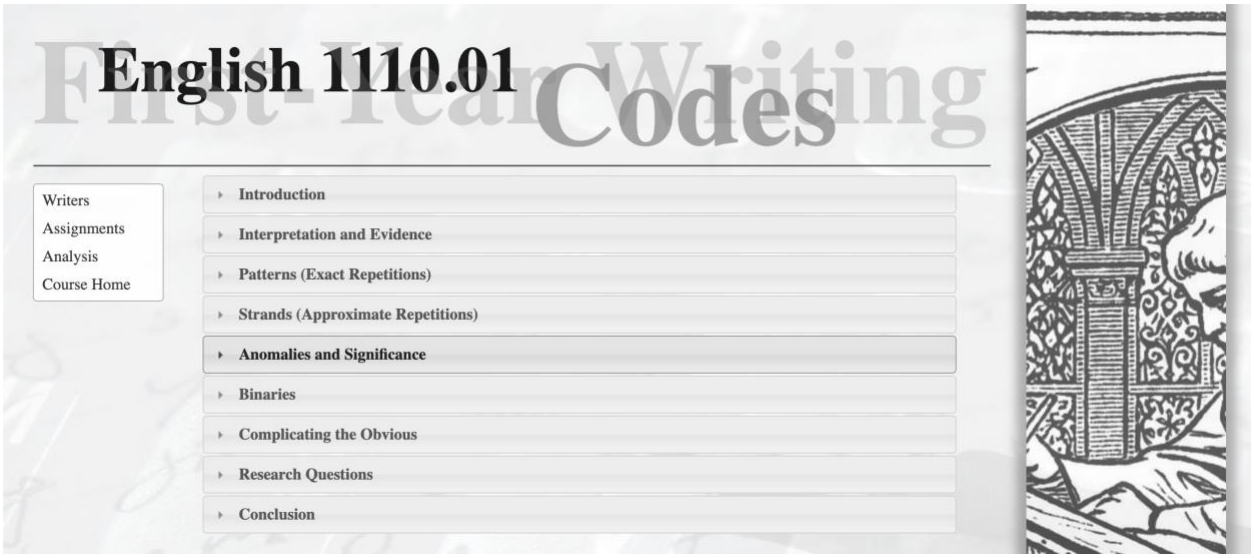


Figure 24: Corpus menu for viewing aspects of the primary source analysis across all students' work

Appendix B: Abbreviated Syllabus for HON 301-004 Honors Proseminar

Critical Editing and Publishing in the Digital Humanities

Spring 2019, Lewis Honors College, University of Kentucky

(boilerplate and course policies have been excised in favor of sections that establish the goals, nature, and organization of the course)

Description

Why code in a humanities class? What do we learn about archival materials when we transform them into digital objects? How do editors profoundly shape a reader's experience of both the information and its interface? These questions are of the utmost importance in the "digital humanities," a wide array of scholarly activity that involves the use of digital technologies in the service of humanistic inquiry. For this course, we'll focus on one of the more storied traditions in digital humanities: the production of an electronic edition that curates and takes a critical approach to a collection of textual artifacts from the University's Special Collections archive.

Work on this edition, "Letters from Devastation: Mary Breckinridge in the Aisne, 1919," began in 2018. An influential figure in the history of nursing and midwifery, Breckinridge founded the Frontier Nursing Service in Kentucky. Specifically, we'll work with personal correspondence and other artifacts when Breckinridge cared for mothers and children in the French countryside in the wake of the First World War. Our team of editors will continue to add more artifacts to the

edition, refine the coding and information design, and deepen the scholarly context in which the edition situates itself. All students will be credited as editors.

This course does not require any previous experience with archival and editorial work, nor does it require any experience with markup language and coding. This will be a team-based and project-based course; thus, the majority of the semester will be spent working together as a team to manage a complex digital humanities project and transform a selection of textual artifacts into a high-quality, critical electronic edition. Along the way, we'll explore questions such as:

- What is the role of the editor in selecting and commenting on source material? How can the editor's work affect the meaning and reception of a text?
- How do methods of digital reproduction, from image scans to coding, enable and constrain our understanding of texts? Moreover, how do they transform those texts?
- How does the multidisciplinary and team-based nature of digital humanities work challenge our traditional notions of scholarship, authorship, intellectual labor, and academic relevance?
- How do archival materials allow us to appreciate and better understand both their historical context and our current moment, and how might we best preserve and publicize those artifacts for a wide audience?
- What are the evolving best practices for encoding textual artifacts and producing electronic editions, and how can things like XML tags and stylesheet transformations profoundly affect the meaning and reception of texts?

Beyond these questions, this course should be of interest to students who want to work with peers across the disciplines in a project-based, collaborative learning environment on materials and information of interest to scholars and community members in Kentucky and beyond.

Learning Outcomes

The curriculum, course content, learning activities, assessments, and projects in this course are all aligned to the following learning outcomes. By the end of the course, we will have:

- contributed positively and equitably in a small, interdisciplinary, team-based, project-driven environment;
- made intentional editorial choices and evaluate the role(s) of the editor in projects that involve the curation, encoding, framing, and presentation of historical documents;
- conducted professional research—both primary and secondary—that provides an audience-centered framework for a scholarly edition of historical documents;
- selected and applied Text Encoding Initiative (TEI) compliant extensible markup language (XML) in the context of a primary document and the purpose/audience of the scholarly edition;
- described the general theories, disciplinary frameworks, methodologies, and practices that inform or represent work in the digital humanities, especially with respect to the work of text encoding and digital scholarly editions.

Required Reading and Other Materials

The required readings and other resources to study will be listed (with links) in the schedule below. You do not need to purchase anything for this course; we use either freely accessible, open resources, or resources to which you have open access as a UK student.

Communication

Because this is a semester-long, project-based endeavor, we will use a productivity app called Slack (it's free!) that allows much more efficient communication than emails. I will send invites to our Slack channel, which will prompt you all to sign up for an account, but you also can register for an account on Slack's website. Part of your graded participation will occur over Slack (i.e., the reading responses), but we will also use it for teamwork, troubleshooting, etc., when we are not face-to-face (and sometimes when we are!). You can join our specific Slack work group by clicking here.

Grades and Assignments

Reading Responses: 50 points

MAC Newsletter Submission: 100 points

Nursing Clio Submission: 100 points[128]

Transcriptional Markup: 100 points

Contextual Markup: 100 points

Campus Presentation: 50 points

Reading Responses. For each day when we have some reading assigned, post at least one reflection (no more than a few sentences) in the Slack channel designated for reading responses. These questions and observations ideally will locate points or ideas that exhibit tension or ambiguity, that challenge or provoke our thinking, that confuse us or seem unclear, or that offer transformative insights for our project. They should serve as a basis for our discussion and be applied in our work. These should be posted to Slack by noon on the day when the reading is due. Feel free to reply within the Slack channel to observations and questions, or prepare thoughts in anticipation of class discussion at or after 2:00.

MAC Newsletter Submission. As part one of a two-part, end-of-semester writing portfolio project, we will collaboratively author an essay critically reflecting on our work for the assistant editor of technology and archives of the Midwest Archives Conference Newsletter. This will involve collectively brainstorming a focus and structure for the project, managing our time and

[128] While the groundwork was laid for the *Nursing Clio* submission via conversations with the site editors at an NEH Institute on Advanced Topics in the Digital Humanities, the assignment ultimately was abandoned over workload and bandwidth concerns, especially to ensure progress on the transcriptional and contextual markup. The spring 2018 section of the course featured a midterm assignment that asked students to write a critical review and give an oral presentation on an existing digital documentary edition, e.g., from the *Romantic Circles* electronic editions site at https://romantic-circles.org/editions.

responsibilities together, drafting roughly 250-300 words per person, and revising the draft collectively to get it up to standards for professional consideration.

Nursing Clio Submission. As part two of the culminating reflection on your work this semester, you will write a brief, polished essay informed by your research and editorial work, designed for the publication Nursing Clio. This writing will be sophisticated and research-informed, building upon a central message designed for a lay audience interested in medical humanities, medical history, nursing, and women's studies/history. These will be submitted to the publisher for review at the end of April, and the editorial staff will select any that they'd like to publish on their website. I have arranged this with the NC editorial staff and they are eagerly awaiting your work.

Transcriptional Markup. Each student will be responsible for selecting documents, digitizing them, transcribing them, and encoding them with TEI-compliant XML transcriptional markup. Moreover, the transcriptional markup ought to adhere to the specific scheme that we will collectively choose for our work. The transcriptional markup will attend to the material, structural, and linguistic content of the writing.

Contextual Markup. In addition to transcriptional markup, we will also apply contextual markup to the documents we have encoded. Contextual markup includes named entities (people, places, organizations, events, etc.) and editorial annotations (explanatory, interpretive, biographical,

historical, linguistic, etc.). Contextual markup in some cases requires extensive and/or creative approaches to researching items of interest in the documents.

Campus Presentation. At the end of the semester, we will present to a broad, campus audience on the work we've done to make UK's archival holdings more accessible to the public, and to shed more light on a critical figure in Kentucky history. This presentation will ask that each team member participate equally, and that visuals be used in some way throughout the presentation.

Schedule

Date	**Topic**	**Readings**	**Notes**
1/10	Code, Models, and Digital Humanities	Kirschembaum: “Hello Worlds”	Register for Slack
1/15	Editorial Work and Digital Editions	Schreibman: “Digital Scholarly Editing”	Complete course intake form
1/17	Mary Breckinridge and Devastated France	Goan: Introduction; Campbell: “Mary Breckinridge and the American Committee for Devastated France”	
1/22	Archives and Digitization	SAA: archives definitions and types; Jimerson: “Embracing the Power of Archives”	Meet in Special Collections lobby with student ID
1/24	Archival Research and Mary Breckinridge	Goan: pp. 30-57; Breckinridge: Selected Letters from 1919	Meet in SCRC Research Room
1/29	Archival Research and Mary Breckinridge	Breckinridge: *Wide Neighborhoods*, ch. 9, 10, 11	Meet in SCRC Research Room

1/31	Archival Research and Editing	Williams and Abbot: *Bibliographical and Textual Studies* pp. 1-8; 12-14; 71-89	Meet in SCRC Research Room
2/5	Archival Research and Digital Editing	Huitfeldt: "Markup Technology and Textual Scholarship"	Meet in SCRC Research Room
2/7	Archival Research and Digitization	Thompson: "Why Don't Archivists Digitize Everything?"	Meet in SCRC Research Room
2/12	Digital Scholarly Editing and Text Encoding; Visit with Dr. Melanie Goan	Pichler and Bruvik: "Separating Encoding from Presentation"	Meet in SCRC Research Room
2/14	XML and the Text Encoding Initiative	Burnard: *What is the Text Encoding Initiative?*	**Due: digitized document files** Meet in SCRC Research Room
2/19	Digital Editing and Correspondence	Hankins: "Correspondence: Theory, Practice, and Horizons"	Meet in classroom Download <oXygen/> and register for a trial
2/21	Metadata and Encoding Correspondence	Stadler, Illetschko, and Seifert: "Towards a Model for Encoding Correspondence"	**Due: complete <teiHeader> for your documents**
2/26	Transcriptional Markup		
2/28	Transcriptional Markup		
3/5	Transcriptional Marlup		
3/7	The "Digital" and Scholarly Editions		**Due: transcriptional markup**
Spring Vacation			
3/19	Contextual Markup		
3/21	Editorial Annotations	Bauer and Zirker: "Literary Annotation and Digital	

		Humanities”	
3/26	MAC Newsletter Planning Contextual Markup and Editorial Annotations		
3/28	Contextual Markup and Editorial Annotations		
4/2	Designing the Edition	Example Editions	
4/4	Designing the Edition		
4/9	MAC Newsletter Workshop		**Due: full draft of MACN submission**
4/11	Preparing the Presentation		
4/16	Preparing the Presentation		**Due: full draft of presentation with materials**
4/17	*No Class: Friday*		**Due: contextual markup**
4/18	Campus Presentation		Meet in SCRC Great Hall
4/23	Debrief and Workshop MAC Newsletter Submission		
4/25	Digital Texts and Humanities	Flanders: “The Literary, the Humanistic, the Digital”; Klein, “Code”	**Due: MAC Newsletter final draft**
5/1	*Celebration at Kentucky Native Cafe (optional)*		*Rain location TBD*

Appendix C: Digital Edition Written Review and Discussion Assignment Prompt

Overview

In 2 to 3 single spaced pages with descriptive headers that respond to the questions that follow, review one of the electronic editions from *Romantic Circles*, a scholarly website and publisher devoted to English literature, culture, and history during the late 18th and early 19th centuries. Choose one of the following suggestions or browse the collection and propose your own choice to me as soon as possible (I need to approve all choices not on this list). Ideally, everyone will choose a different edition, but if two of you want to focus on the same one you can coordinate the presentation as a group.

- *William Wordsworth's Guide to the Lakes*, edited by Paul Westover, et al.
- *The Collected Letters of Robert Southey Volume 1*, edited by Lynda Pratt
- *The Letters of Robert Bloomfield and His Circle*, edited by Tim Fulford and Lynda Pratt
- *Sporting Sketches During a Short Stay in Hindustan*, edited by Tilar Mazzeo
- *A Letter to the Women of England*, edited by Adriana Craciun, et al.
- *Lyrical Ballads*, edited by Bruce Graver and Ron Tetreault
- *New Letters from Charles Brown to Joseph Severn*, edited by Grant Scott and Sue Brown

Components

Using descriptive headers and chunking information into well-formed paragraphs, your review must *in some way* address all of the following questions. This does not necessarily mean that each question will constitute an independent section of your review. Some questions can be

answered in a few statements, while others will need considerable space for exploration. Or, you might choose to organize your review based on the content of the edition as opposed to the sequence of questions. Depending on the edition you've chosen, some questions will require more elaboration or attention than others. What matters is that you address them all in *some* way.

- What kind of text does the electronic edition feature? In other words, what is the rationale of the edition? Is it thematic, work(s)-based, author(s)-based, or some other kind of content? Does it feature a wide array of texts, or does it focus on a very specific set of texts? How, if at all, implicitly and/or explicitly, does the edition embed itself and make a case for itself in the critical or scholarly discourse on its artifact(s) or topic(s)? How does the edition exert agency, intervene in, or contribute to the scholarly discourse? (For example, the edition of the relatively obscure long poem *Thoughts in Prison* by William Dodd presents a text with which most readers would be unfamiliar as a way of better understanding a heretofore unappreciated influence on a much more well-known poet and poem, Samuel Taylor Coleridge's "This Lime-Tree Bower My Prison.")
- What seems to be the overall purpose, goal, rationale, and audience for the edition? Do the editors explicitly state this, and if so, where and how? If the editors do not explicitly address this question, does the edition implicitly answer it in other ways? What type of an edition is this (e.g., documentary, genetic, critical)?
- Does there seem to be an intended way that readers are expected to interact with the edition? How is that intention conveyed, and how might there be alternate ways that the edition could have been designed from user experience (i.e., how the reader can, is encouraged to, and desires to interact with the edition) and textual/editorial theory

perspectives (i.e., theories of what constitutes a "text" and an "edition" as well as the type and degree to which editors may exert influence on the text and the reader)?

- What components (e.g., kinds of appendices, additional media, prefatory matter) have the editors chosen to include, and how much do these components take advantage of the abilities of an electronic/digital medium? Would these components be possible in a print edition, and if so, how would they work differently from both editorial and reader perspectives? What other components (or different versions of the components already present) might the editors have included, especially considering the possibilities that markup language and digital processing afford? How would they change or bolster the edition's *ethos*, purpose, user experience, etc.?
- How does the edition (re-)present the textual artifact(s) that it features? What sort of information is added, and what sort of information is subtracted from the audience's experience of the texts? How does the presentation of the featured text(s) align (or not) with the edition's purpose/rationale, as well as with the possibilities afforded by markup language and digital processing for web browsers? What are some alternate ways that the featured text(s) could have been presented, and how would those change the effect, purpose, or *ethos* of the edition? What seems to be the implicit theory of the text (or, even, textual ontology) that underwrites how the edition presents its featured text(s)?
- How much do the editors make the audience aware of the editorial theories and practices that were used to compose the edition? What information about the edition are we given, and what does that information say about the nature of (digital) editorial work? What information was not included, and how might that be useful for certain readers to know?

- In responding to any of the questions above, how can you ground your responses in the scholarship, theories, methods, language, and discourse that we've studied thus far (e.g., on digital editing, on markup language, on digital humanities)?

Class Presentation

In addition to submitting the written review via Canvas for assessment purposes, please share your document on Slack for us to have on hand during your presentation. For this assignment, "presentation" signifies a conversational adaptation of the insights in the written review for an *informal* conversation during which you tell us about the edition you reviewed and highlight the most important insights from your review. During the conversation you should use the academic language of editing, electronic editions, and digital humanities as much as possible (this is when the class glossary comes in handy). You may also want to prepare a few images for us to review, or to direct us to navigate through particular sections of the electronic edition. We all should feel welcome to ask questions during and after these informal presentations/discussions which should last at least 10 minutes but no longer than 15 minutes.

Assessment Criteria

This is a *formative* assessment; that is, its main purpose is to help us consolidate our learning so far and look ahead to later work in the class. It should feel—and is—a low-stakes assignment. As such, the following criteria will be used to determine grades. Overall a successful written review and presentation will help us better understand the possibilities of digital scholarly editing and move us towards more clarity regarding our work for the edition of Breckinridge's letters.

- Use of (and facility with) the language and theories of editing, electronic/digital editions, and digital humanities;
- A good faith attempt to address the prompt questions in some way, with special focus on those that are most relevant to the edition at hand;
- Articulation of take-aways, insights, questions, problems, or ideas for our work on the edition of Breckinridge's 1919 correspondence and associated artifacts;
- Conversational command of your own written work and ideas on the edition;
- Clarity of written mechanics, language, and organization of ideas.