ABSTRACT

Title of Document: HACKING LITERATURE: READING ANALOG TEXTS IN A DIGITAL AGE

Aaron Dinin, Doctor of Philosophy, 2014

Directed By: Professor Martha Nell Smith, Department of English

Evangelists of the digital age, in the immediacy of its adolescence, often describe digital technologies as "revolutionary" (e.g. "the digital revolution") and as having a world-changing impact on human cultural interactions. However, by considering digital media from a temporally scaled vantage point spanning thousands of years, *Hacking Literature* proposes ways in which the digital age might also be introducing "world-saming" technologies that are as likely to reinstantiate cultural norms as they are to create new ones. *Hacking Literature* finds evidence for its arguments by considering examples of similar technological innovations prevalent in "revolutionary" technologies of information storage and dissemination: that of differently mediated literary texts. Using arguably iconic examples from Homer, Shakespeare, Eliot, and Dickinson (an epic, a drama, a novel, and poetry), and creating analogies between those texts and, respectively, the Linux kernel, Internet security protocols, the history of the World Wide Web, and the world's most successful

blogging engine, *Hacking Literature* describes ways in which literary media and digital media appear to undergo similar kinds of technological transformations. The project then analyzes these similarities to suggest possible opportunities for using software development concepts as entry points for literary analysis, as critical lenses for reading that meld technology and humanities.

HACKING LITERATURE: READING ANALOG TEXTS IN A DIGITAL AGE

By

Aaron Dinin

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park, in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2014

Advisory Committee:
Professor Martha Nell Smith, Chair
Professor Marilee Lindemann
Professor Jason Rudy
Professor Melanie Kill
Professor Ben Bederson

Table of Contents

Introduction: Connecting Technologies

The following project is the result of my uncommon combination of professions. One of those professions – the one directly spurring this project – is the study of English literary texts and the surrounding scholarship. The second profession is the building of complex web software applications. In other words, I am both an English doctoral student working, learning, and collaborating in an academic setting, and I am a software engineer working, learning, and collaborating in a corporate setting. I acknowledge my work in two professions rarely associated with each other because, despite claims to the contrary, the two fields do not often collaborate, or at least not in the ways I will emphasize. I do not mean that people in academic humanities disciplines do not know how to code software, since people like Matt Kirschenbaum, Steven Ramsay, Adeline Koh, Natalia Cecire, and Franco Moretti, among many others, regularly work in, discuss, and build code-based projects. Instead, in nearly a decade of work within the two professions, I rarely encounter other people who are similarly committed to working in the corporate world of software development. Instead, often when I explain my background, I encounter responses of surprise. People either wonder why a developer is also pursuing a PhD in English, or they wonder why someone pursuing a PhD in English also works as a developer.

The answer, my experience has taught me, is that the knowledge used in the two fields is not nearly as dissimilar as might at first appear. Specifically, the knowledge I have acquired as a developer has made me a

better reader of literary texts, and the skills cultivated while analyzing literary texts have helped me become a better developer. At its simplest, this project expands on these overlaps. But their existence also suggests a somewhat overlooked relationship between the two professions and the opportunities available in encouraging more collaboration of the kinds described here and more sharing of knowledge.

Perhaps the most appropriate way for me to begin exploring the relationship between software development and academic English studies is to start with the usual explanation I give when asked why I study both software and literature. Language – I like to remind those who ask – makes use of technologies that help users create, store, and disseminate information. Thousands of years ago, in a time well before written and oral histories, our ancestors were doing what successful species must do: they were developing more efficient ways to survive. The difficult task of survival is easier thanks to the technologies evident in language. Among other things, language helps packs of hunters better coordinate their strategies. Language helps communities preserve and transmit learned information across generations. Language helps one person convey to another person important warnings such as, “Watch out for the pointy thing on that animal’s head,” which limits the number of times communities need to learn fatal lessons. In other words, in these examples language serves as a survival tool and, in that regard, might be considered a technology.

Humans have been using language for so long that the majority of the world’s population is capable of creating, storing, and disseminating

information by customizing linguistic structures with a relatively high level of complexity. Our comfort deploying language has become so complete that, even though many people are aware that language is artificial, most will rarely contemplate language during their day-to-day usage of it as anything but natural and "human." In contrast, software-based mechanisms of information storage and dissemination are so new that only a small portion of the population knows how to code custom programmatic structures, and most software users regard programming languages as artificial and unnatural. But the act of coding can serve a purpose similar to the act of writing: as tools that help store and disseminate information.

When given this context, not only does the chasm between literary scholarship and software development shrink, opportunities emerge that have not been recognized as widely as would be beneficial to both fields. Since coding and writing can enable similar social purposes, studying literary texts should be able to contribute to a knowledge of coding, and studying programmatic software should be able to contribute to a knowledge of writing. Because of this relationship, my explanation for why I both code and study literary texts is this: Knowing how to code informs and improves my ability to study literature, and studying literature informs and improves my ability to code digital applications.

What follows in the coming chapters are a series of examples showing deployments of software and literature suggesting the kinds of developmental, structural, and cultural similarities that might allow knowledge of each discipline to inform and improve study of the other. The project's goal is to

leverage those examined similarities in order to suggest the kinds of critical and procedural methodologies that such comparisons might allow. Because the primary audience of this project is not the software development community (though I encourage them to read it), I will not focus on possible ways in which the study of literary texts can improve one's ability to code. Instead, this project will present similarities between literary texts and software development concepts in order to demonstrate opportunities for literary analysis.

Appreciating the potential of software development concepts in relation to the study of literary objects can be difficult when limiting the temporal context for the study to the relatively small timescape of the digital age. However, "Scal[ing] enlargement along the temporal axis," as Wai Chee Dimock encourages in *Through Other Continents: American Literature Across Deep Time*:

> Changes our very sense of the connectedness among human beings. It also suggests that different investigative contexts might need different time frames, with no single one serving as an all-purpose metric. Some historical phenomena need large-scale analysis. They need hundreds, thousands, or even billions of years to be recognized for what they are: phenomena constituted by their temporal extension, with a genealogy much longer than the life span of any biological individual, and interesting for just that reason. A shorter time frame would have

> cut them off in midstream, would have obscured the fact of their cumulation.[1]

Dimock is arguing that an appropriate understanding of some phenomena is impossible within the context of the relatively short contextual spans in which most people normally study them (what she describes as a "biological" life span). Instead, they require a temporal frame of analysis that can span centuries or millennia.

Hacking Literature wonders if an example of one such historical phenomenon that can benefit from Dimock's concept of "deep time" analysis are the evolutions of literature in different media. By expanding the temporal scale for considering information technologies from the miniscule immediacy of current digital trends into the much larger scale of literary storage and transmission technologies, digital technologies can be more thoroughly studied, appreciated, and leveraged as tools of analysis. Instead of studying technological fads likely to become little more than cautionary tales or cultural punch lines within a decade,[2] contextualizing digital media within a temporal scale spanning civilizations can augment the sense that digital technologies are just as likely to be revolutionarily world "saming" as they are to be revolutionarily world changing. Recognizing that new technologies – in whatever forms they arrive – are as capable of reinstantiating cultural

---

[1] Dimock, Wai Chee. *Through Other Continents: American Literature Across Deep Time*. Princeton, NJ: Princeton University Press, 2008. L 87

[2] As N. Katherine Hayles notes in the prologue of *My Mother Was a Computer: Digital Subjects and Literary Texts*, the evolutionary pace of digital technologies can make serious critical discussion of popular digital objects a shortsighted exercise (1-11, esp. 2). Even as I write this document, I find myself wondering which of my statements about digital technologies will, within a few years, seem dated and obsolete.

priorities as making new ones emphasizes an additional analytical approach to literary studies in the digital age that seems underrepresented in current methodologies of digital-critical synthesis.

Current deployments of digital technologies in relation to non-digital media tends to emphasize the use of digital tools to support textual analysis. For example, critics like Stephen Ramsay use digital tools to process information more quickly and analyze large data sets.[3] Another common emphasis of digital technologies in the humanities is the using of digital media to provide better access to information through processes such as data mining, information retrieval, and archival presentation.[4] Although these kinds of uses are helpful analytical and critical resources, they also imply the sole value of digital technologies are as tools to enable more efficient iterations of already existing processes. For example, technically, Ramsay could produce the kinds of analysis he creates through his concept of algorithmic criticism without computers. Computers just make the process easier and faster.

But the digital age, when recognized as yet another milestone on a technology development continuum that spans millennia instead of centuries, has an additional analytical benefit. Specifically, witnessing the earliest, rapid

---

[3] Ramsay's proposed "algorithmic criticism" concept praises computers for their power of scale. He asserts that, "The computer revolutionizes, not because it proposes an alternative to the basic hermeneutical procedure, but because it reimagines that procedure at new scales, with new speeds, and among new sets of conditions" (*Reading Machines*, 31).

[4] Although early digital humanities work emphasized building digital tools as though it fulfilled a fantasy of "universal access," the field, as a whole, seems to have recognized their limitations. As Ramsay points out, "It is manifestly impossible to read everything, and it has always been so. The utility of the digital corpus – despite its vaunted claims of 'increased access' – only serves to make the impossibility of comprehensive reading more apparent (though a stroll through the stacks of even a modest library serves to illustrate the same point)" (78).

evolutions of new digital technologies provides otherwise difficult-to-uncover information about the ways in which early technologies develop to become more stable and more successful tools for cultural information storage and dissemination. What does paralleling the evolutionary processes of digital technologies with similar structures in older technologies of information storage and dissemination reveal about the ways in which information technologies develop? What does imposing that understanding on the past suggest about the ways in which prominent users of older information technologies both deployed and improved those technologies to help people better store and transmit information across generations?

Because I emphasize an analytical approach that frames millennia instead of shorter literary periods, I recognize how some of my statements and my project's scope might seem grand. Concerning this largeness of scope, I should first note that I do not imagine an audience of specialists. While each of the primary, chapter-level examples I present throughout the project could be considered in significantly more detail, the general purpose is to introduce lenses for reading texts that help to better contextualize pre-digital textual production for digitally-native audiences. As a result, at certain moments throughout the project, I take what, to a specialist, might seem to be descriptive shortcuts that are intellectually problematic. In these moments, I provide relevant footnotes, where appropriate, in order to direct readers to more detailed discussions.

Beyond targeting a non-specialist audience, I also view this project's seemingly large scope as providing better context for the analysis I am

suggesting. I deploy a large temporal frame in order to denote commonness across generations, cultures, and civilizations. Although a frame of "across civilizations" might, itself, seem large, in the context of evolving technologies of information storage and dissemination, that frame is relatively small. Even a technology as established as handwriting, for example, is, in a larger context of human history, relatively new.[5] In this sense, my framing is akin to zooming out on an image. By zooming out, we have less detail, but better context with which to more adequately judge proportions. As the previous observation from Dimock explains, the same benefit is true of scaling a temporal timeframe.

What follows in the coming pages is an approach to digital humanities[6] that emphasizes the "digital" only to the point that readers should be able to easily recognize the digital theme of the discussion in terms of its relationship with other similarly influential media for information storage and dissemination: epic; drama; prose; poetry.

Hacking Literature's first chapter continues the work of this introduction by better positioning the cultural applications of digital technologies in relation to the cultural applications of non-digital technologies. Grounded with a thesis that technologies of information storage

[5] The oldest records of handwriting date to approximately 3200 BCE, but the oldest known human remains are estimated to be around 400,000 years old.

[6] I apply the phrase "digital humanities" only because it is has become a common title for liberal arts theory discussing digital technologies, not because I think the label is appropriate. It can be, as the chapters in this project will hopefully imply, misleading because it implies a more profound separation between digital technologies and humanist pursuits than this project is willing to accept.

and dissemination – whether digital, literary, cultural, biological, or otherwise – serve the same fundamental goal of preserving information across generations, the chapter shows that literary technologies are often acting as tools for storing information and providing more efficient information access.

The first chapter's framing of information technologies as responses to a cultural need for preservation and transmission of shared knowledge leans heavily on Derrida's concept of "archive fever." From within that frame and using a Walt Whitman poem as an example, the chapter is able to provide a basic demonstration of how digital technologies and literary technologies often parallel one another in their data storage and retrieval processes. The seemingly simple example introduces the potential that software has an analytical framework through which its users can supplement their study of literature. It shows how the analytical value of software extends beyond using it as a tool for data processing or a tool for presenting texts on a screen. The patterns used by software applications to connect their users with remotely stored information are also indicative of some of the ways in which literary technologies fulfill the same societal purpose.

After introducing and framing the existence of procedural concurrencies between software development and writing, the project's remaining four chapters further demonstrate the potential for the analytical value of software evolutionary and developmental processes by unpacking a series of increasingly more complex technological analogies. I present the analogies in parallel with each other within the chapters in order to give each

chapter a dual center based around the juxtaposition of one literary and one digital technology.

The first of these analogies is the relationship between the Linux operating system and Homeric epic. The development of both technologies have, since their initial releases, progressed in a parallel fashion. One of the significant differences, aside from digital-ness or lack thereof, is the timescape during which the evolutions have occurred. We can round the age of Homeric epic in millennia; in contrast, we have to round age of Linux in decades. But the chapter demonstrates how the relative brevity of Linux's existence helps empower its role as a critical tool. The comparison begins by mapping both Homer and Linux to similar developmental trajectories. But the chapter is less interested in directly comparing Homer and Linux and more interested in demonstrating a general overlap in order to gesture at other similarities, perhaps not as obvious, but that have valuable analytical potential during further study of the Homeric texts.

By engaging with the controversy and uncertainty of the Homeric tradition, the second chapter, in addition to its primary purpose, also serves as useful preparation for the third chapter. The third chapter introduces into the project another literary name similarly surrounded by authority, controversy, and uncertainty: William Shakespeare. By having already treated Homer as a complex idea rather than a single person, the project positions itself to include a discussion of Shakespeare that, while not directly engaging with it, recognizes that the Shakespeare about which I am writing is less of an

individual and more of an idea, and exemplifies the author function Michel Foucault theorizes.[7]

Because so much has been written about Shakespeare, featuring him (and the idea of him) in a single chapter might appear to be, if my goal were comprehensiveness, a problematic decision. One chapter is not enough space to adequately address the many ways in which both the concept of Shakespeare as well as the texts attributed to him operate as technologies of information storage and dissemination. But my goal, again, is not comprehensiveness. My goal, instead, is to leverage an understanding about how digital technologies evolve in order to explore how and why the name William Shakespeare – particularly when used in the context of a project like this – introduces so much cultural context that even including it might overwhelm anything else in the chapter or those chapters surrounding it. The Shakespeare name, as Foucault theorizes and the chapter considers in relation to Internet security, has become a sort of security mechanism that helps protect the information stored within the stories on which he bases his plays.

While chapters on Homer and Shakespeare create analogies between technologies that have no direct relationship with each other, the fourth chapter, using George Eliot's *Middlemarch* as its thematic foundation, pairs a literary technology with a digital technology it helped inspire. It compares the technology of the novel with the World Wide Web in order to understand how the latter technology was developed at least partially in response to the

[7] Foucault, Michel. "What is an Author?" *Language, Counter-memory, Practice: Selected Essays and Interviews*. ed. Donald F. Bouchard. Ithaca: Cornell University Press, 1980. 113-138.

perceived shortcomings of the former. The goal of the chapter is not to position one technology as superior to the other, but is to show that no technology is without its limitations. In so doing, the chapter argues that the most skilled users of any technology are the ones best able to overcome those limitations through creative and artistic innovation.

To underscore this link between technology and artistic innovation, the fifth chapter explores the underlying artistic medium of digital technologies: software engineering. Instead of discussing software engineering through the lens of "computer science," this final chapter examines code through a lens of poetry and does so by analyzing the programmatic structures within the poetry of Emily Dickinson. The chapter features Dickinson's poetry paralleled with programmatic language not because the metaphors of my project require it, but because Emily Dickinson could be considered, as the chapter hopes to demonstrate, a software engineer. She uses a type of code to program the world's most complex kind of processing device: the human mind.

Hacking Literature touches a range of literary and critical texts that span thousands of years while navigating subjects that are, themselves, capable of occupying lifetimes of study. This kind of textual wayfaring could appear capricious if underlying decisions for this project's subject matter and the rationale behind those decisions remain unacknowledged and unexplained, and this introduction concludes with some reflections on those choices.

The decision to incorporate texts and authors not often analyzed alongside one another is related to the value I have found, while developing my technological knowledge, in pushing against traditional topical frames such as profession, genre, geopolitical boundaries, and historical moment. Such frames are valuable for categorization and analysis within a given era or discipline, but they can limit understandings of a larger historical and temporal window in which information technologies evolve. Since one of my goals is to frame digital information technologies as representing relatively minor iterations of prior technologies in an evolutionary phenomenon that spans thousands of years, *Hacking Literature's* argument benefits from emphasizing similarities rather than differences in epic, poetry, prose, drama, and software.

Jay Clayton, in *Charles Dickens in Cyberspace*, suggests that deemphasizing traditional critical boundaries such as, in his example, genre, also results in "Preserving an acute sense of the different functions of genres."[8] In my case, I have learned a great deal about the different functions of traditional critical boundaries of, among other things, topic, time period, genre, and critical specialty. Among those lessons learned, the most relevant to this project has been recognizing their value as, themselves, key operating mechanisms for improving the technologies of literature. Some examples of the importance of these traditional disciplinary and conceptual boundaries within technologies of literature that have become more apparent to me include their ability to provide subject-specific concepts with which readers

[8] Clayton, Jay. *Charles Dickens in Cyberspace*. New York, NY: Oxford University Press, 2003. 28.

will be familiar, standardizing preferred textual structures, and defining at least loose boundaries for relevant critical material.

Not having some of these standard boundaries in place will demonstrate limitations and opportunities within the technologies of literature, print, and digital media. As Clayton argues, overwriting traditional genres "interrogates the nature of such boundaries, asks what work these categories perform, why they change over time, how they organize aspects of experience far removed from the aesthetic."[9] The same hopefully will be said of this project. By mixing epic, software, religion, novels, coding, drama, Internet security protocols, poetry, the French Renaissance, and more, and by including Shakespeare, Linux, Homer, Dickinson, Tim Berners-Lee, George Eliot, and others, I demonstrate ways in which the boundaries between those topics can limit their study. Concomitantly, I can reinforce the value of their boundaries.[10]

I have selected the specific authors around which center four of my chapters – Homer, Shakespeare, Eliot, and Dickinson – for two primary reasons. First, the prominence of these authors allows me to position them as partial surrogates for their peers and so could suggest that similar arguments might be made in relation to any number of other authors and works. However, because including all such examples would be impractical, by opting for recognizability over obscurity, I am attempting to deploy a kind of

---

[9] *Ibid.*

[10] Although I emphasize this project's attempt to disregard some standard textual boundaries, I should also note that, despite its diverse subject matter, the project is by no means free of influential structural frameworks. Chief among them is the framework of "English dissertation," which influences much of its structure, its style, and its content.

intertextual linking I discuss later in the project in order to suggest that my examples are illustrative, not comprehensive.[11]

The second reason for my selection of the primary literary figures in this project relates to the ways in which *Hacking Literature's* primary authors intertextually link to each other as they build their texts on top of the technologies of those who preceded them. Though Homer cannot directly link to those who follow him, Shakespeare directly links to Homer, Eliot directly links to Homer and Shakespeare, and Dickinson directly links to Homer, Shakespeare, and Eliot. These authors were already using each other's texts as cultural technologies of information storage and dissemination. Putting them in conversation within this project explores information technologies through some of the connections its focal authors have already made.

[11] See Chapter 4.

Chapter 1: I View the Body Electric

In 1979, a Norwegian computer scientist named Trygve Reenskaug invoked a metaphor to construct an organizational principle for computer software programs. Reenskaug's metaphor was so well received that its concepts remain one of the dominant architectural patterns for many of today's most sophisticated online software applications. Since the underlying code for generating such applications is, for people unfamiliar with software development, often difficult to understand, I will explain Reenskaug's contribution to software development without invoking a single line of code. Instead, I will rely on the familiar work of unpacking metaphors.

Before beginning that process, I should note a shortcoming of attempting a code-less approach to describing software architecture. Once a programming language is learned – a task less difficult than most people assume – understanding the functionality within an application's codebase is often easier than understanding its structuring metaphors because metaphoric interpretation is not processed consistently. Instead, the success of metaphoric interpretation, as a feature of literary technology, depends on knowing the data within the inaccessible minds of one's audience.

A similar task of connecting with inaccessible minds is what spurred Trygve Reenskaug's software development metaphor during a year in which

he worked as a visiting scientist with the Smalltalk[1] research group at Xerox's Palo Alto Research Center (PARC). At Xerox PARC, Reenskaug was attempting to theorize a "general solution to the problem of users controlling a large and complex data set."[2] In other words, Reenskaug was trying to define a way of organizing information to make accessing and manipulating the data within a software application easier for the application's users. The kinds of software he was considering are known as graphical user interfaces (GUIs).

In the early 1970s, the research being done at Xerox PARC was laying the foundation for the GUIs through which most users would learn to interact with their digital devices. The metaphors we often take for granted on everything from our desktop computers to our cellular phones and ATM machines – things like windows, icons, buttons, and menus – were, in the 1970s, only first being imagined. The purpose of developing those metaphors was to enable users to manipulate a computer's data import and export processes in productive ways while eliminating the need to work with those processes directly through computer code. For example, the task of saving a document for modern computer users usually involves little more than activating some sort of 'save' icon or selecting a 'save' option from a software menu or keyboard shortcut. But these simple save metaphors are visual representations that trigger a computer's programmatic and usually unseen

[1] Smalltalk is a programming language created at Xerox PARC in the 1970s primarily by Alan Kay, Dan Ingalls, Adele Goldberg, Ted Kaehler, and Scott Wallace. The main purpose of Smalltalk and its many derivatives is as an educational and theoretical language for concept mapping.
[2] Reenskaug, Trygve. "MVC XEROX PARC 1978-79." n.p. n.d. Web. 8 Feb. 2013. http://heim.ifi.uio.no/~trygver/themes/mvc/mvc-index.html.

codebase. Interacting with the visual ‘save’ icons initiates the actual programmatic command processes needing to be completed in order to accomplish the ‘save document’ task.

As the work of building user interfaces became more complex, developers encountered a production bottleneck. All the code being used to create GUI elements (e.g. things like save buttons) were being built into the software alongside the code needed to accomplish their associated programmatic processes. The result was user interface code mixing with application code. This mixing made little difference to the end users of software applications, but for the programmers, mixed code with different purposes created logistical problems.[3]

Reenskaug's metaphor, on its surface, provides a solution for separating a program’s user interface code from its application code, or what is often referred to today as a software's "business logic."[4] His solution

[3] If I were being precise, an argument could be made that individual file length could technically be slowing down processes and, as a result, slowing the user experience. While I will not be so exact here, or in other similar explanations throughout this documents, I should at least note that, at certain professional levels of programming, the fractions of seconds used to execute lines of code are important when considering code structures and file sizes. However, the primary issues incited by mixing interface code with operational code are usually realized in large applications. As is the case in most complex systems, the larger the system, the more unwieldy it is to alter. When a program consists only of a couple thousand lines of code written by a single developer within a short period of time, the mixed code is not a major obstacle. But when software scales to hundreds of thousands of lines of code written over the course of years by multiple people, organization is critical for successful iteration.

[4] Business logic, sometimes referred to as “domain logic,” usually consists of the functions for accessing data and interfacing with the machine on which a program is running in order to determine what information to output. In contrast, the user interface code determines how that information should appear on an output device such as a monitor or printer.

appropriates a metaphoric pattern he called Models-Views-Controllers, a concept more commonly referenced by its initialism: MVC.[5]

The MVC pattern is not a programming language, nor is it a piece of software, nor is it a specific application of any kind. MVC is a structural metaphor for organizing a complex software application's internal codebase. The metaphor has been applied to programs ranging from operating systems to video games, but its most popular use, at the time of this writing, is as the organizational principle for complex web applications.[6]

In the MVC pattern, "models" include all of an application's data access logic. Reenskaug specifically defines the concept of a model as, "an active representation of an abstraction in the form of data in a computing system."[7] For the sake of clarity, I will more simply define a model as the portion of a software's codebase that enables the following processes: creating new information to store within an application's dataset; retrieving information from an application's stored data; updating existing information in the application's dataset; or deleting obsolete, outdated, or no-longer necessary

---

[5] Models-Views-Controllers was Reenskaug's second, slightly revised version of his metaphor. His first version, published eight months earlier, was called "Thing-Model-View-Editor." The changes from his first iteration of the concept to the second were mostly nominal. The "Thing" component of the concept was a reference to the overall project being worked on, and the basic "Editor" components were attributed to the newer "Controller" concept, while the remaining concept of the "Editor" was redefined as "an ephemeral component that the View creates on demand as an interface between the View and the input devices such as mouse and keyboard" (Reenskaug, "MVC XEROX PARC 1978-79").

[6] "Popular" is a difficult-to-quantify term. Here it references number of unique software programs using the pattern. While the MVC architectural paradigm is used in a variety of programs, the number of websites relying on MVC, particularly when compared with the relatively limited number of programs on the scale of, for example, operating systems, is the factor I am using to generate a sense of relative popularity.

[7] Reenskaug, Trygve. "Thing-Model-View-Editor: and Example from a planningsystem." 12 May 1979. PDF File. http://heim.ifi.uio.no/~trygver/1979/mvc-1/1979-05-MVC.pdf.

application information.[8] Even more simply (though somewhat errantly), a model can be described as the code that interacts with the database.

Reenskaug's explanation for the "views" portion of the MVC pattern is less technical, and, as a result, likely more easily understood. Reenskaug explains that: "A view is a (visual) representation of its model. It would ordinarily highlight certain attributes of the model and suppress others. It is thus acting as a *presentation filter*" [emphasis in the original].[9] A helpful way of thinking about Reenskaug's concept of a view is to imagine it as the part of an application responsible for producing the visual representation users are able to see and manipulate primarily on screens. An application's views are meant to contain all the visual markup attributes that enable our graphical user interfaces.

The third component of the MVC pattern is the concept that, for people new to the metaphor, often creates the largest conceptual challenge. While programmatic models equate directly to manipulating an application's stored data, and programmatic views create the visual interface, programmatic "controllers" have no direct correlation to an easily visualized object.

The difficulties inherent in visualizing a controller have led to a common misrepresentation of the MVC paradigm that errantly depicts a controller as a sort of in-between link in a metaphoric MVC triad. This errant depiction suggests that controllers enable communication between what the user is seeing and interacting with (a software's views) and the processes

[8] The programmatic process of creating, retrieving, updating, and deleting data is often referred to by its acronym: CRUD.
[9] Reenskaug, Trygve. "Models - Views - Controllers." 10 Dec. 1979. PDF File. http://heim.ifi.uio.no/~trygver/1979/mvc-2/1979-12-MVC.pdf.

designed to access an application's storage mechanisms (the software's models). This faulty explanation is so prevalent that many available explanations of the MVC pattern, specifically as they relate to web application development, position a website's controllers as the intermediaries between models and views. For example, the following visual representation of the MVC triad comes directly from an explanation of MVC on the website of the Oracle Corporation:

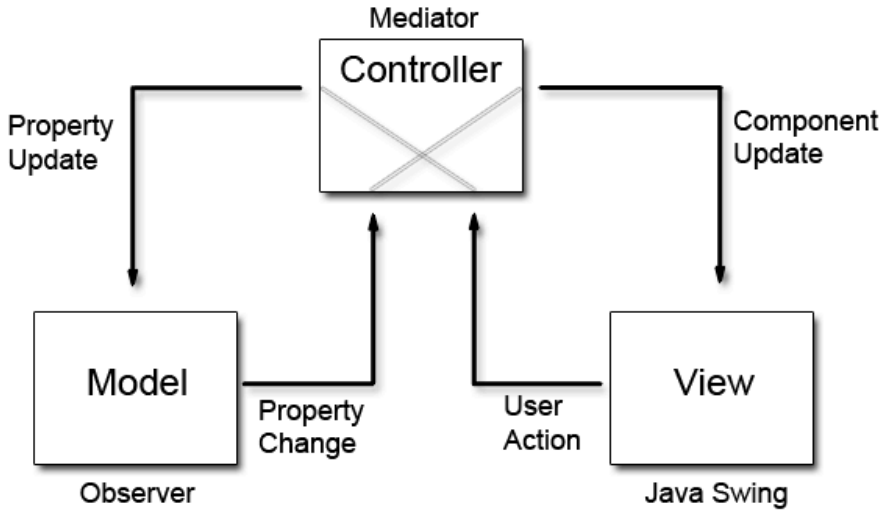

Figure 1: A visual representation of the MVC software architecture paradigm provided by the Oracle Corporation.[10]

According to Reenskaug's original conception of the process, this understanding of a controller as an intermediary between an application's user interface and its information abstraction layer is flawed. Controllers are not meant to operate as intermediaries between models and views. Instead,

[10] Eckstein, Robert. "Java SE Application Design With MVC." Oracel. Mar. 2007. Web. 10 Jul. 2012. http://www.oracle.com/technetwork/articles/javase/mvc-136693.html.

controllers are intended to provide what Reenskaug describes as "the link between a user and the system."[11]

The distinction between Reenskaug's intended purpose of controllers within a piece of software and the common misconception of their purpose is well-visualized by a representation of the MVC pattern appearing on Reenskaug's personal website. In Reenskaug's depiction, the controller and view are visually joined together, demonstrating that the controller is not an intermediary between view and model, but instead, the controller works with the view to act as an intermediary between the human user and the computer model:

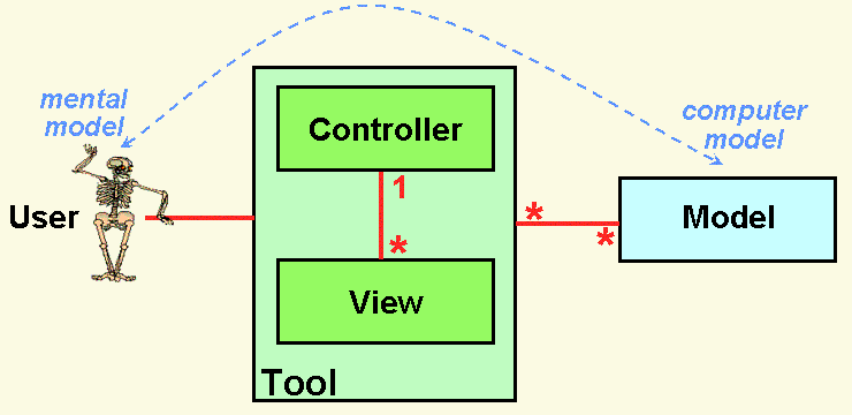

Figure 2: A visual representation of the MVC paradigm as imagined by Trygve Reenskaug, the metaphor's original creator.[12]

In one of the most common examples of a user interacting with MVC-patterned software, when a user visits a website using MVC, the website's controllers can process everything about the user's visit, including the kind of computer being used (PC, phone, tablet, etc.), the web browser being used, whether or not the user has already been to the site, whether the user has a

[11] Reenskaug. "Models - Views - Controllers."
[12] Reenskaug. "MVC XEROX PARC 1978-79."

currently active registration session, as well as a number of other traceable user-dependent variables. From that information, the controller then determines what views to display, and, in the process, delivers to the user the most appropriate version of the website.

As I previously suggested, metaphors do not always simplify what they describe, and in the case of Reenskaug's MVC metaphor, not only is the concept difficult to grasp for most non-programmers (as well as, I should note, many programmers), it even complicates the code itself. MVC-patterned applications are often more complex than purely procedural software,[13] they usually require more code to accomplish the same work, and the additional structuring hierarchies often burden computer processors and networks with heavier processing loads.

Despite the MVC pattern's complexities, by the end of the first decade of the 21st century, MVC had become the most popular architectural pattern for the development of complex websites and web applications. Coinciding with this rise in popularity of MVC patterned web development practices is a period in the Internet's history called Web 2.0. Are these parallel rises to prominence related?

Before attempting to answer that question, let me provide some context for Web 2.0. What distinguishes Web 2.0 from the Web's prior incarnations are the ways in which users interact with the content and functionality of websites. In a pre Web 2.0 website, content is more akin to

[13] The procedural paradigm is one in which a program operates by executing a series of often, though not always, linear step.

the content in books, magazines, and newspapers, meaning site visitors interact through the site only passively, primarily by reading. As web technologies evolved, new methodologies were developed which enabled users to not just read content, but also to create their own content. Web 2.0 represents the ensuing rise to prominence of websites where users could directly engage with content by means of creating their own.

Because I am writing this document in the middle (or possibly end) of the Web 2.0 age, I consider myself unqualified to consider the significance of the Web 2.0 technological moment through any reliable historically contextualized lens. However, even though I lack a temporally-removed historical perspective on the Web 2.0 age and its eventual cultural significance, I have the advantage of a first-hand perspective on its immediate cultural impact. That cultural impact seems aptly described by Alan Liu as "the Big Bang explosion of new communication forms centered on the ability of users to participate in the creation, sharing, and linking of content."[14] From within the programmatic and cultural detritus caused by Liu's "Big Bang explosion of new communication forms," we can see the following parallel ascension: Web 2.0 websites have become prominent in the cultural mainstream and Reenskaug's 30-year-old structural metaphor has become the primary architecture behind those websites. Why is Reenskaug's metaphor so well suited to undergird the structural and cultural phenomena particular to Web 2.0's productivities? What does the MVC metaphor, despite

[14] Liu, Alan. "Friending the Past: The Sense of History and Social Computing." *New Literary History*. 42.1 (2011): 1-30.

its complexities, enable in web software architecture that makes it the preferred architectural pattern for Web 2.0 software development? And is the MVC metaphor particularly conducive to technologies that facilitate user inner-connectivity and interaction?

MVC itself does not directly provide tools for user interactivity on websites, meaning Reenskaug's concept is neither a Web 2.0 catalyst nor a necessary Web 2.0 component. This relationship, or lack of a relationship, is underscored by the common usage of the MVC metaphor decades prior to the emergence of Web 2.0, meaning MVC and Web 2.0 have a non-causal relationship. Neither one directly produced the other, so their relationship is not fully explained by a sense of technological necessity. Beyond technological compatibility, the two programmatic constructs also seem to be connected by their conceptual compatibilities. Specifically, MVC's underlying data access metaphors and Web 2.0's information access metaphors function in complimentary ways. But, despite their mutual relationship with digital media, those complimentary ways are not technology-driven; they are purpose-driven. That shared purpose is more easily recognized when MVC and Web 2.0 are removed from the temporal constructs of "the digital age" and recontextualized within a broader sense of cultural information storage and transmission technologies that also include literary technologies represented by things like epics, dramas, novels, and poems.

The first clue that MVC is a concept with significance beyond website architectures comes from Reenskaug himself. According to the engineer's

original note outlining what would become the MVC pattern, Reenskaug derived his metaphor by considering the ways in which most large projects are accomplished. Instead of immediately appending his metaphor to exclusively digital technologies, Reenskaug begins his explanation by describing the MVC concept in relation to creating a general object: a "thing." A "thing," according to Reenskaug's own definition of the term, is: "Something that is of interest to the user. It could be concrete, like a house or an integrated circuit. It could be abstract, like a new idea of opinions about a paper. It could be a whole, like a computer, or a part, like a circuit element."[15] The "thing" concept Reenskaug describes, and which the MVC pattern enables, can be anything. He never limits the end products to software applications or computer code.

After providing his explanation of a "thing," Reenskaug gives an example of a thing that has no relationship to software. Instead, he uses the example of "A Large Project," suggesting possible projects as "the design and construction of a major bridge, a power station or an off-shore oil production platform."[16] He then explains that these kinds of large projects are accomplished by packaging them into individual tasks. To describe this separation of tasks, he uses an abstraction model popular in the 1970's called "activity networks." The basic principle of an activity network is to define each task needing completion in order to successfully accomplish a project and then demonstrate which tasks need to be done in what order so they can then be assigned to and handled by their proper facilitators.

[15] Reenskaug. "Thing-Model-View-Editor: and Example from a planningsystem."
[16] Ibid.

Because the end goal of the MVC pattern is to accomplish an efficient structure for creating a "thing" as opposed to a complex piece of software, Reenskaug's methodologies are organization-driven, not topic-driven. Reenskaug does not approach software development as a unique task requiring uniquely defined processes. Instead, Reenskaug is making the case that a complex software application should be viewed as, fundamentally, nothing more than a large project, and, like any large project – be it coding software, building a skyscraper, or even writing a dissertation – successfully accomplishing it requires a systematic and purposeful division of processes.

Reenskaug begins to define this distinction when he explains the purpose of MVC, writing that "The essential purpose of MVC is to bridge the gap between the human user's mental model and the digital model that exists in the computer."[17] For Reenskaug, MVC's essential purpose has nothing to do with software creation. Instead, MVC is about "bridging a gap" between the human user and the data stored within an application's database.

Upon initial consideration, bridging a gap between humans and digital models might not seem like a concept far removed from a surface-level understanding of MVC that ties it to computer software production methodologies. However, Reenskaug's final, succinct definition for a Model, presented in a second note to his Xerox PARC team a few months after his first, reveals otherwise. Reenskaug explains that: "Models represent knowledge."[18] He never limits his concept of knowledge to computers, or

[17] Reenskaug. "MVC XEROX PARC 1978-79."
[18] Reenskaug. "Models - Views - Controllers."

software, or anything digital. Even if Reenksaug's conception of models might be digital representations of knowledge, the knowledge itself is, presumably the same basic concept of knowledge that most humanists, philosophers, artists, poets, painters, musicians, composers, academics, and everyone else have been pursuing, so far as anyone can tell, since the beginning of recorded history, and probably long before.

By recognizing that Reenskaug's vision for Models is for them to be digital representations of knowledge, we can uncouple his MVC paradigm from software application development. Instead, understanding Models as digital representations of knowledge reveals how the MVC metaphor helps address what Reenskaug refers to as the "larger project" – a project he recognizes as the need to bridge the gap between humans and stored knowledge. Understood from this perspective of connecting humans with knowledge, MVC appears less like a structuring system for a technical apparatus and more like an organizational system undergirding a humanist pursuit. Books bridge the gap between humans and knowledge. Paintings bridge the gap between humans and knowledge. Epic, poetry, music, drama, film, and photography all help bridge the gap between humans and knowledge. Since the MVC paradigm is the dominant metaphor on which many of Web 2.0's most prominent online applications are built, is it possible a website is a humanist pursuit? Are websites bridging the gap between humans and knowledge?

Although not always explicitly stated, the subtext to every chapter in this project is to wonder about precisely these kinds of questions concerning

digital software. Software development, because it relates to computers, is regularly and often unquestioningly lumped into a broad and indiscriminate category known as "computer science." This categorization has happened despite the fact that the primary task of a software developer's work is the implementation of language. While the study and development of language as it relates to linguistic phenomena might indeed be a scientific pursuit, the implementation of language, particularly within fluid medias of knowledge dissemination through abstract and metaphoric representation, has long been a literary and humanist endeavor.

I do not intend for this demarcation between what might be called "scientific work" and "humanist representations" to be as stringent as it might at first appear. I am not encouraging readers to immediately (or eventually) disconnect the work of computer coding from the scientific foundations of digital technologies. Nor am I prepared to argue for the inclusion of fine exemplars of programmatic composition into the cannons of literary masterpieces. Instead, I ask anyone inclined to resist the notion that software development could be an artistic endeavor to first consider both programmatic and literary processes not from a perspective of historical pragmatism, but instead to interpret them through a lens of romanticism.

By suggesting romanticism, I do not specifically mean Romanticism with its capital *R*. While I would encourage the Romantic era's willingness to push against political and social boundaries, I do not suggest the work should be performed emotionally or somehow in opposition to scientific rationality. Instead, I embrace a willingness to idealize some of the purposes served by

both literary texts and digital software. Doing so reveals far more similarities than what is normally expected when comparing scientific processes and literary texts, and it provides users of both technologies additional approaches for considering their relationship with each other. Though perhaps we should not be too surprised. Albert Einstein, a pillar of scientific discourse, reminds us that, "The fairest thing we can experience is the mysterious. It is the fundamental emotion which stands at the cradle of true art and true science."[19] And Walt Whitman, one of America's most recognizable poetic figures, tells us "In the beauty of poems are the tuft and final applause of science."[20]

As both Einstein and Whitman hint, a willingness to consider the science of software and the art of literature without limitations of traditional academic and professional boundaries produces not just passing similarities, but an abundance of intersections. The abundance is so great that one limited text like this project could not begin to fully articulate the ways in which software and literature overlap, intermingle, and inform one another. As a result, I will, by necessity of space and knowledge,[21] limit myself to one

[19] Einstein, Albert. *The World As I See It*. Minneapolis: Filiquarian Publishing, 2006. 14-15.
[20] Whitman, Walt. "Preface 1855 – *Leaves of Grass*, First Edition." *Leaves of Grass and Other Writings*. Ed. Michael Moon. New York: W.W. Norton & Company, Inc., 2002. 626.
[21] The limitation of knowledge is perhaps the more significant component than limited space because software programming, like writing, is not one static concept or discipline. Rhetoricians David R. Russell and Arturo Yañez provide a helpful metaphor to describe the complexities of writing texts that can be also be appropriated for writing software. They explain: "A lot of games are played with a ball, just as a lot of fields use the tool called writing. But the ball is different, the rules of the game are different, the object of the game is different. And knowing how to shoot a basketball (or write in one way, one genre) doesn't mean you know how to throw a baseball (or write in a different activity or genre). Learning a new game (or academic field and its ways/genres of writing) means participating in a new activity, and using tools (including the tool

intersection – an intersection that seems to have a strong presence in Reenskaug's MVC metaphor and its need to bridge a gap between humans and knowledge.

Bridging a gap between humans and knowledge is, in its own right, a difficult concept to interpret because it, too, is a metaphor. People have knowledge. There is no gap. While some of it is biologically inherent – like breathing – most of that knowledge is acquired through learning. Learning, here, is not meant to be limited to an academic sense of the word, but instead should be understood as that knowledge which is acquired through experiencing and responding to the world's stimuli. The "gap" needing to be bridged, as Reenskaug is identifying in his metaphor, is not necessarily the acquisition of knowledge through learning (a skill most people are readily capable of, if not always interested in implementing), but instead he is referring to the manipulation of knowledge belonging to some sort of external 'other.' This understanding of the metaphor can be teased out by returning, briefly, to the concept of a digital model. A digital model's purpose is to manipulate, through programmatic processes, real-world information (i.e. knowledge) contained within the storage mechanisms of a digital surrogate. The bridging enabled by the MVC pattern does not provide direct access to all the world's vast repositories of knowledge. Instead, MVC bridges a person's internal processes for manipulating knowledge (what Reenskaug calls "the

---

or technology we call writing) in different ways" (336). A similar explanation can be extended to software development. Although, in this project, I tend to reference software development as a generally similar discipline no matter the kind of software, the reality is significantly more complex. My knowledge of those complexities, like my knowledge of the complexities of the literary topics in this project, does not even approach comprehensiveness, nor could it.

human user's mental model") with a software's internal processes for manipulating knowledge. This bridge provides a program's user with access to the repository of information – being represented here as "knowledge" – stored within a software application.

Once the secondary "bridging" metaphor is unearthed from Reenskaug's pattern and understood, the purpose of MVC, as well as any web application built using the MVC paradigm, should be more apparent. MVC helps users better input and extract information from an external, stored repository of knowledge – an archive. As a result, the development of MVC software no longer needs to be understood as only the foreign work of computer science and unintelligible (to many people) programming languages. Instead, MVC can be recognized as the much more familiar work of archival interaction.

Any comfort one might derive from viewing software development as a process for archival interaction is fleeting; archiving and archival interaction are, themselves, problematic processes. Archives are, in fact, so problematic that I often wonder if the hardest part of becoming a good software engineer may not be learning to code or learning how certain technologies work. Instead, the challenges of writing software might be the same challenges encountered during any critical pursuit. They could be the challenges of archive and what Jacques Derrida calls "Archive Fever."

Archive fever, as presented in Derrida's book of the same title, is not, as some common readings imply,[22] overly concerned with the physical space of archive. Archive fever is about self-preservation in the face of the absolute certainty of destruction. Specifically, Derrida explains that:

> There would indeed be no archive desire without the radical finitude, without the possibility of a forgetfulness which does not limit itself to repression. Above all, and this is the most serious, beyond or within this simple limit called finiteness or finitude, there is no archive fever without the threat of this death drive, this aggression and destruction drive.[23]

For Derrida, archive fever is a response to what Sigmund Freud calls the "death drive." Although the name *death drive* implies otherwise, the concept is not directly tied to physical death. Freud alternately refers to the death drive as the "aggression drive" and the "destruction drive," and its primary concern is the inevitable disappearance of everything.[24]

---

[22] Ed Folsom makes this implication when he writes: "Any of us who have spent time in actual nineteenth-century archives know the literal truth of Jacques Derrida's phrase 'archive fever.' As Carolyn Steedman has argued, real archives may well produce something pathological in the researcher that might be named archive fever, because archives reify the period they record. They contain not only the records of a period but its artifacts as well, their dust the debris of toxins and chemicals and disease that went into making the paper and glue and inks, that went into processing the animal skins that wrap the books we open and, in the dusty light, read and inhale. When we emerge from an archive, we are physically and mentally altered. We emerge with notes – photocopies if we're allowed – but never with the archive, which remains behind, isolated from us. Archive, if a genre, is one that only a few ever read. Archive fever demands narrative as an antidote, and many of our books (and virtually all our biographies) are tales of archive survival" ("Database as Genre." 1577).

[23] Derrida, Jacques. *Archive Fever*. Chicago: The University of Chicago Press, 1996. 19.

[24] More information about Freud's concept of the death drive can be found in *Beyond the Pleasure Principle*.

As both Derrida and Freud assert, nothing lasts forever. This ultimate death of everything includes memory. Archive fever is, according to Derrida, the human response to the mortality of memory. He explains that:

> [T]he archive, if this word or this figure can be stabilized so as to take on a signification, will never be either memory or anamnesis as spontaneous, alive and internal experience. On the contrary: the archive takes place at the place of originary and structural breakdown of the said memory.
>
> *There is no archive without a place of consignation, without a technique of repetition, and without a certain exteriority. No archive without outside.* [emphasis in the original][25]

Archive, in the Derridian sense, serves as a sort of cultural coping mechanism. It helps preserve and transfer memory that would otherwise be lost. And archive fever, in addition to being a response to the dissipation of memory, can also be understood as a recognition of the futility of archive itself. No matter how extreme or meticulous people are in their archiving endeavors, the simple principles of ultimate destruction inherent in Freud's death drive ensure that even the best made archives will not last.

Archivization's ultimate failings are not enough to deter the human pursuit of archive. As a result, Derrida decides to classify such futile pursuits as a sort of terminal disease. I want to add an additional classification. Although Derrida associates the futility of archivization with a human

[25] Derrida. *Archive Fever*. 11

unwillingness to acknowledge mortality, I see the unwavering certainty of Freud's death drive as possibly assigning the act of archiving a dual purpose. Instead of prioritizing archive as a process to preserve memory for the sake of an implied cultural immortality, I propose an understanding of archive that also acknowledges it as a source of past information best leveraged to inform future decisions. Archive, to return to Reenskaug's MVC metaphor, is the external repository of socio-cultural knowledge for which humans need specialized tools in order to gain access.

In the context of other historical technologies of information communication, the ability of digital technologies to enable archive is certainly not unique; however, digital technologies have dramatically advanced archival capabilities of information storage and information access.[26] These advances have resulted in (and are continuing to result in) what might best be described as "hyper-archives." As my naming convention implies, hyper-archives signify the human propensity for archivization realized in the context of a digitally fueled environment providing unprecedented resources of storage and distribution.

In his book *Delete: The Virtues of Forgetting in a Digital Age*, Viktor Mayer-Schönberger describes the result of this transition toward hyper-archivization, explaining that:

[26] Matthew Kirschenbaum's *Mechanisms: New Media and the Forensic Imagination* is a good resource to begin examining the ways in which digital technologies contribute to information storage and access.

> Since the beginning of time, for us humans, forgetting has been the norm and remembering the exception. Because of digital technology and global networks, however, this balance has shifted. Today, with the help of widespread technology, forgetting has become the exception, and remembering the default.[27]

The cultural implications of what Mayer-Schönberger describes as a move away from forgetting and toward remembering is discussed at length in his book.[28] More important to this project are the reasons behind that cultural shift. The shift toward remembering discussed by Mayer-Schönberger is the byproduct of hyper-archiving.

In contrast with Derrida's suggestion that archiving is a struggle with the death drive, the current rapid progress toward a culture of remembering seems to have relatively little interest in preserving information for the sake of delaying our shared cultural mortality. Consider, for example, that the primary tools of digital archivization are overwhelmingly commercial. The most recognizable of these is Google, the digital behemoth that began as a search engine for finding information within the digital archive more commonly known as the World Wide Web. Since its launch as an archival extraction tool for the Web, Google has evolved into a vast archive of

---

[27] Mayer-Schonberger, Viktor. *Delete: The Virtue of Forgetting in the Digital Age.* Princeton, NJ: Princeton University Press, 2009. Location 84.

[28] The quote cited here ends with the explanation that: "How and why this happened, what the potential consequences are for us individually and for our society, and what – if anything – we can do about it, is the focus of this book" (Location 84).

information about its users.[29] That user archive is so large that many operators of websites are often less concerned with being on the Web and more concerned with how their websites will be positioned specifically to access Google's archive of users.[30] But Google's varied deployments of archives are largely unconcerned with preserving the past for the sake of delaying the inevitable disappearance of everything. Instead, Google's archives are designed specifically to better connect its users to knowledge. This goal is clearly articulated by the company's mission statement, which famously explains that: "Google's mission is to organize the world's information and make it universally accessible and useful."[31]

Similar to Google, another popular digital archive of information that dwarfs any non-digital archive is Facebook. Instead of specifically describing how Facebook preserves a rich history of cultural interaction – a process that should be self-evident to anyone remotely familiar with the platform – I will suggest a brief thought experiment. Imagine anachronistically importing the kind of archival record of human communication and interaction Facebook provides into contentiously debated moments in human history. For example,

---

[29] Mayer-Schönberger describes Google's archive of interactions well when he explains that: "Google remembers search queries, even though a billion such requests reach it every day. If somebody watches you surf the Web, after a few days that person will have forgotten what you searched for and when. Google will not; its memory of one's search requests is much more durable, as well as accurate" (Location 1392).

[30] The positioning of a website in a Google search (or any search engine's query) is commonly referred to as its "rank." Because people's search habits rarely take them beyond the first page of results when performing search queries, a website's success (and, in many cases, the success of the company running the website) can often depend on its ability to be ranked on the first page. As a result, a multi-billion-dollar industry has formed around helping websites rank highly in search engines using a principle called Search Engine Optimization (SEO). One of the most successful companies in the SEO industry, as well as one of the best resources for more information about it, is SEOmoz (http://www.seomoz.org).

[31] "Company." *Google*. Google, Inc. Web. Feb 9, 2013. http://www.google.com/about/company/

would questions about religious icons be nearly as prevalent (or controversial) if Jesus and his apostles had been writing on each other's Facebook walls? Alan Liu alludes to how a website like Facebook has the ability to redefine the process of historical preservation when he explains that:

> The difference of Web 2.0, of course, is that the whole calculus for converting spatial separation into temporal narration once constitutive of the sense of history has been reconfigured. In the age of digital networks, spatial barriers to sociality seem fewer and less impassable. This is due to technological progress. But it is also due in great part precisely to the fact that the dirtiest work of war, pacification, oppression, and modern empire building has already been done to create a world safe for the transport of both capital and information. Web 2.0, in other words, is a libertarian pygmy standing on the shoulders of a tyrant ogre. As a result, spatial-political barriers that once took muscular civilizations centuries, if not millennia, to traverse by pushing through roads, etc., are now overleaped in milliseconds by a single finger pushing "Send."[32]

While Liu's consideration of Web 2.0 and social media's impact on historicity and, implicitly, archive are not necessarily positive,[33] the recognition of its impact underscores the potential Web 2.0 applications could have as agents of cultural preservation.

---

[32] Liu, 22.

[33] Liu concludes his essay by personifying social media and offering it a sort of cultural warning, explaining, "My argument is that the amplest experience of sociality includes the society that is history, and social media will be more fully human if it remembers that" (Liu, 25).

Despite the immense power to preserve human memory via archive enabled by digital technologies, few people seem concerned with using their digital tools as a means of protecting human culture from inevitable destruction.[34] Not that we should be surprised. In retrospect, the same scarcity of interest in current cultural preservation is endlessly looped throughout history, when few people were using the new tools of previous milestone technologies, like writing and printing, specifically for the purpose of archiving. Instead, the archives of digital technology are, like the archives of so many technologies to come before the digital age, either the products, the tools, or the byproducts of the then-current culture's primary means of communication.

But perhaps I am being too inclusive with my classification of what constitutes an archive. Perhaps, in contrast to the commercial digital repositories of information in which I see archive, Derrida's sense of archives are as repositories more self-conscious about their processes and their intentions of cultural preservation. Derrida even makes specific mention of the difficulty in determining where archive begins, writing that:

> Because it is entrusted to the outside, to an *external* substrate and not, as the sign of the covenant in circumcision, to an *intimate* mark, *right on* the so-called body proper. But where does the outside commence? This question is the question of the

[34] Few does not mean "none." One notable example of preserving, in particular, web culture, is *The Internet Archive* (http://www.archive.org).

> archive. There are undoubtedly no others. [emphasis in the original][35]

This confusion about where "outside" and archive commence admittedly destabilizes my sense of what an archive is and, as a result, my ability to definitively define the work of sites like Google and Facebook as archives. Perhaps because their primary objectives are not preservation and archivization, they should not be considered archives.

I am not alone in my inability to clearly define what is and is not an archive, a point Derrida makes when he describes the struggles anyone considering the topic ultimately encounters. He explains that:

> [C]oncerning the archive, Freud never managed to form anything that deserves to be called a concept. Neither have we, by the way. We have no concept, only an impression, a series of impressions associated with a word. To the rigor of the *concept*, I am opposing here the vagueness or the open imprecision, the relative indetermination of such a *notion*. "Archive" is only a *notion*, an impression associated with a word and for which, together with Freud, we do not have a concept. We only have an impression, an insistent impression through the unstable feeling of a shifting figure, of a schema, or of an in-finite or indefinite process. [emphasis in the original][36]

[35] Derrida, 8.
[36] Derrida, 29.

Despite the instability of what Derrida is describing as the "notion" of archive, I still find in Derrida's assertions a definite sense of general agreement. The "we" of Derrida (ourselves and the author, though Freud should be included as well) might not have a clear and perfectly articulated definition of archive, but this lack of specificity highlights what might be, for a discussion of connecting humans with knowledge, the most important point. The communal, culturally shared sense of archive seems to understand archive as a repository of preserved information, and information provides the component pieces of knowledge. Despite how someone might choose to define the constraints of what may or may not be included within archival or archive-worthy information, and despite how someone might choose to define the constraints for who may organize such repositories, the presence of information – the existence of knowledge – remains unchanged. And archives, just like the data in software applications, remain repositories of knowledge in need of tools to bridge the gap between preserved archival knowledge and the people wanting to extract it.

On their surface, digital advances in the tools of archive that increase the human ability to preserve knowledge offer significant cultural benefits. Among those benefits, digital technologies appear to be benefiting the size and scope of archives themselves. But the increased scale of archiving has revealed a flaw in the tools used for extracting information directly from archives.

A literary metaphor from Jorge Luis Borges artfully unravels the digitally magnified flaws of increased archive capacity. In the Borges short story "The Library of Babel," the author introduces a fictional library "composed of an indefinite and perhaps infinite number of hexagonal galleries."[37] Each of these galleries contains an identical number of variations of a single 410 page book, and the library itself, throughout its hexagonal galleries, is said to contain every possible 410 page variation. The numbers equate to at least $1.956 \times 10^{1,834,097}$ books.[38] Throughout these seemingly endless rooms of books, the inhabitants of Borges' library try to find the one book, a "catalogue of catalogues,"[39] that can make sense of the library. However, the vastness of the library makes the likelihood of anyone finding the One Book so remote, it never happens.

Borges's "Library of Babel" parallels what digital technologies might be doing to archives. Easily overshadowed in the excitement about the benefits to archiving enabled by digital technologies, whether as an academic pursuit or as a commercial pursuit, is the limitation the same technology has yet to fully overcome. The limitation is, in a sense, an iteration of the limitation Derrida contends with in *Archive Fever* – it is the limitation of time. And the limitation of time is a result of mortality. The ability to connect people to increasingly vast stores of knowledge is of little use when we consider the inability of people to ever sufficiently navigate them.

---

[37] Borges, Jorge Luis. *The Library Of Babel*. Trans. J.E.I. Web. 5 Oct. 2011. http://jubal.westnet.com/hyperdiscordia/library_of_babel.html.
[38] Bloch, William Goldbloom. *The Unimaginable Mathematics of Borges' Library of Babel.* Oxford University Press: Oxford, 2008.
[39] Borges. *The Library of Babel*.

The vastness of information provided by the digital age underscores a trend made more apparent by many of the significant innovations in information storage and dissemination technologies. This trend is well-illustrated by Bertran Gervais, who, while questioning the "text" of a screen, explains that:

> Manuscript culture corresponds to what historians of literacy and reading refer to as an intensive reading situation (where few texts are read, but they play an essential role in the life of the reader), while book culture corresponds to an extensive reading situation (where many different texts are read, but in a superficial manner). In manuscript culture, books are important and of a religious nature, while they become cultural goods in book culture. In our linked computer culture, texts are simply overflowing. It is a context of hyperconsumption of cultural goods, which the terms browsing, surfing, or even navigating especially evoke.[40]

As Gervais explains, while new technologies increase access to mechanisms of knowledge storage and dissemination, users of those new technologies tend to adapt to the additional knowledge available by devaluing it. The result of the current increase of knowledge-containing objects produced by digital technologies has led to a culture of hyperconsumption in which the knowledge people have access to is often devalued because people have far too much

[40] Gervais, Bertrand. "Is There a Text on This Screen? Reading in an Era of Hypertextuality." *A Companion to Digital Literary Studies*. New York, NY: Wiley-Blackwell, 2008. 7.

additional information to filter through and nowhere near enough time to comprehensively filter it. In the vastness of the information repositories enabled by digital media, access to knowledge (for those digitally networked) is no longer a major challenge, nor is storing knowledge. Instead, overcoming problems of information access and information storage has revealed a more challenging underlying problem: how do people connect to the right knowledge?

This question is, quite literally, a multi-billion dollar question. A company like Google, for example, is devoted to finding solutions to it, with its goal being to always have the first result of any digital query (whether in its search engine, mapping system, video network, etc.) be the result that solves the enquirer's needs.[41] Similarly, IBM is in development of its Watson project, an artificial intelligence computer system that "applies advanced data management and analytics to natural language in order to uncover a single, reliable insight."[42] The goal of companies like Google, IBM, and any number of other technology titans and startups, is to parse the ever-increasing repositories of stored knowledge in order to connect people to the knowledge they seek.

The work of companies generating complex algorithms and applications to parse all the world's available information in an attempt to find the best answer might be a solution to the digital age's Library-of-Babel-

[41] This explanation of Google's services is not meant to characterize the company as necessarily noble or altruistic. Part of what defines the "right" results in a Google search query is the money the company receives from advertisers in order to present certain query results in particular orders.

[42] "IBM Watson." *IBM*. IBM, Inc. Web. 9 Feb. 2013. http://www-03.ibm.com/innovation/us/watson/what-is-watson/the-next-grand-challenge.html.

like problem. Indeed, while a person has limited time and speed with which to parse all the world's information – our digital Library of Babel – the speed at which a computer can transverse data could conceivably make finding nearly any solution possible. However, an approach to connecting people with knowledge by improving archival search mechanisms makes problematic assumptions. Among those assumptions, the approach assumes people always know what they are looking for. It also assumes people will recognize the correct solutions when presented with them. And, most egregious of the potential assumptions is to assume a correct answer always exists. While the above assumptions might be true in some attempts of information discovery, are they true in all cases?

In our current system of cultural knowledge transmission through connection to repositories of stored information (i.e. archives), the flaws revealed by advances in digital technology are hardly fatal. Arguing otherwise might imply previous knowledge transmission and extraction techniques from smaller and less robust data stores were somehow perfect solutions, which was certainly not true. For example, even during the golden age of the book as a cultural knowledge repository, parsing a single percentage of all available texts take countless lifetimes. Thus, when put into proper perspective, the digital age has not changed the problem of archival information retrieval; it has, as with many other things, magnified its scale.

However, seemingly lost in the discussion of digital media's transformations of the archival process – through expanded access and storage – is the digital age's influence on an alternate way of "bridging the

gap" between humans and knowledge. Whereas companies like Google are focused on connecting people directly to information as a means of knowledge acquisition, the gap between humans and knowledge can also be bridged by connecting people to people.

In the Internet age, communication between large networks of people has been significantly enhanced by the emergence of Web 2.0 and social media paradigms. The rapid growth of these kinds of person-to-person applications provides my first example of this project's primary suggestion that digital technology paradigms can serve as a critical lens with which to analyze older information technologies and unearth similar operational structures.

Historically, digital technologies are unrivaled in their ability to connect people with one-another, but the attention they receive for this benefit is focused on the immediacy of the connection in what Alan Liu describes as "a kind of quantum social wavefront connecting everyone to everyone in a single, shared *now*" [emphasis in the original].[43] Liu is highly critical of the sense of immediacy offered by Web 2.0. "So what happens to the sense of history in Web 2.0?" he rhetorically asks. "First, we should get the easy answer out of the way," he prefaces, as if the negative impact Web 2.0 connectivity has on history is so universally acknowledged, leaving room for argument is unnecessary; "The obvious answer is that Web 2.0 makes history

[43] Liu, 22

disappear because it takes instamatic to a tweet extreme."[44] Liu's assertions about Web 2.0 frame human connectivity as a condition for communication and interaction. In doing so, he seems to neglect the ability of human connectivity to operate as a tool of cultural knowledge preservation and transmission.

Web 2.0's cultural knowledge preservation and transmission mechanisms are rarely overtly identifiable, which is, I suspect, the reason people like Liu might overlook them. The person-to-person connections available in Web 2.0 seldom encourage direct contribution to cultural preservation. But intentionality, in this scenario, is irrelevant. Web 2.0 interactivity does archive and disseminate shared cultural knowledge. Its ability to do so is a product of what Walt Whitman calls "the body electric."

I turn, briefly, to Whitman in order to provide a first demonstration of the ways in which literary texts and digital applications parallel one another and, in so doing, offer instructive opportunities for reciprocal analysis. I realize Whitman's body electric is a poetic conceit – a metaphor – and poetic conceits are rarely the undergirding principle of analytical arguments. However, since much of the work of this chapter has been the unpacking of metaphors with the aid of other metaphors, allow me, first, the opportunity to use Whitman's "body electric" as a critical tool, and second, as a result, a chance to suggest the value of both MVC and Web 2.0 as frameworks for literary analysis.

[44] Ibid. 19

Unearthing a concept of archival access through person-to-person connectivity in the poem that would eventually be titled "I Sing The Body Electric"[45] requires, first, a shift in the poem's common poetic interpretation. The generally accepted understanding of "I Sing The Body Electric" is to interpret it as what Martin Klammer, in a seminal text detailing Whitman's relationship with slavery, describes as a "celebration of the human body as sacred [that] becomes the basis for an egalitarian democracy where the slave and the immigrant belong."[46] The problem with an interpretation of "I Sing The Body Electric" that emphasizes Whitman's democratic and egalitarian vision for African Americans is that this interpretation contradicts substantial prosaic evidence showing Whitman as, if not necessarily a slavery supporter, then certainly not the devout abolitionist champion many people like to retroactively ascribe to his historical persona.[47] Whitman provides an easy solution to this kind of contradiction when he very famously claims, toward the end of "Song of Myself": "Do I contradict myself?/Very well then I contradict myself,/(I am large, I contain multitudes.)"[48] However, as tempting as leveraging Whitman's clever phrase is when explaining away

[45] "I Sing the Body Electric" began as the fifth of 12 poems in Whitman's original 1855 edition of *Leaves of Grass*. At the time it was untitled and unsectioned. In the 1856 edition, it was named "Poem of the Body," and it took its current title and nearly final form by the 1867 edition.
[46] Klammer, Martin. *Whitman, Slavery, and the Emergence of* Leaves of Grass. University Park, PA: The Pennsylvania State University Press, 1995. 141.
[47] Klammer's introduction does a good job of summarizing this historical dichotomy. It begins by offering a column published by Langston Hughes in which the popular African American poet describes Whitman as the "greatest of American poets" due, in large part, to his inclusiveness of blacks in his sense of democracy. In response to this assertion, an English professor by the name of Lorenzo D. Turner wrote to Hughes claiming: "From a careful study of all Whitman's published works I am convinced that he was not a friend of the Negro, and had very few contacts with Negroes, and thought that they were inferior to other human beings" (Klammer, 1-5).
[48] Whitman. "Song of Myself." 1324-1326.

inconsistencies, there is an easier solution. What if Whitman is not contradicting himself?

The evidence of Whitman's opinions on slavery indicate a man who, while not an advocate of slavery, held opinions about both slavery and the rightful place of African Americans in society that were much more complex than his being an entirely committed abolitionist unabashedly in pursuit of complete equality and democracy for all. Whitman was a man engaged with the slavery discussions of his day, and those discussions were not as simple as pro-slavery versus anti-slavery. The arguments included questions about the rights of Northern states to interfere in the cultural and business operations of Southern states, colonization schemes for blacks, and, as seen in Whitman's temperance novel *Franklin Evans*, questions about biracial relationships.[49] In consideration of these kinds of variant shades of complexity encumbering discussions of slavery during the historical moment in which Whitman is writing "I Sing The Body Electric," instead of attaching one definitive argument to the poem, I propose attaching to it the same complexity of the cultural conversations in which it is written.

Easily overlooked in "I Sing The Body Electric" is what happens when the poem is read in the context of its historical moment. The historical link to slavery is the most obvious example, but the emphasis on the slave auction sections near the poem's end overshadow an earlier portion of the text when the historical moment of publication is equally as impactful. Early in the

[49] Klammer's first chapter examines *Franklin Evans* as, in part, a pro-slavery apology narrative.

poem, Whitman describes a man who was "over eighty years old."[50] For a current reader, the man's age likely signifies both being old and perhaps the kinds of wisdom that come with age, but when the poem is first published in 1855, a man over eighty years old would have been alive both during Whitman's moment of publication, when questions about slavery are pushing the country closer and closer toward civil war, and in 1776, when the country was in the midst of another sort of civil war. Unlike the War Between the States less than a century later, the civil war of 1776 is not remembered as a civil war because the outcome has created the memory of a successful revolution.

The link between Whitman's poem and the American Revolution created by the age of the poem's old man is confirmed by a haphazard annotation in one of the poet's manuscripts. In a document featuring a two-columned list of human body parts that would eventually become the foundation for the final body parts catalogue in "I Sing The Body Electric," Whitman scribbles an instructive mathematical calculation. Between the manuscript's two columns, Whitman subtracts 1776, the year of American independence, from 1856, roughly the time of his writing,[51] which results in

[50] In the 1855 edition, the old man appears on line 33. By the final, deathbed edition, he appears on line 38.

[51] The man over 80 years of age is present in the 1855 version, which causes a slight discrepancy between the 1856 year used in the marginal calculation and the year in which the 80 year result was first available in the poem. A number of explanations can account for this discrepancy, including the possibility that the two-columned list of body parts was actually created prior to the 1855 edition's completion despite its content not being included until the 1856 edition. This argument is supported by common mathematical education practices during the time that encouraged rounding of numbers in order to make mathematical calculations easier (such as rounding 1855 up to 1856 in order to simplify the "1856-1776" calculation). Whitman, while an editor at the *Brooklyn Daily Eagle*, even recommends a mathematics textbook called *Practical Arithmetic* that specifically teaches such calculations, writing: "To teachers who have felt the

the number 80 and, if not a direct calculation to establish the age of the poem's old man, a strong indication that the age of the old man and the American Revolution are related.

want of good text books, (as what teacher has not?) we think we can conscientiously recommend the *Practical Arithmetic*, prepared by James B. Thomson, and published by Mark H. Newman, 199 Broadway, N.Y. It needs but an examination and trial of its merits, to make itself its best recommendation. Can it not be put in our Brooklyn Schools?" (Florence Freedman, 152-153).

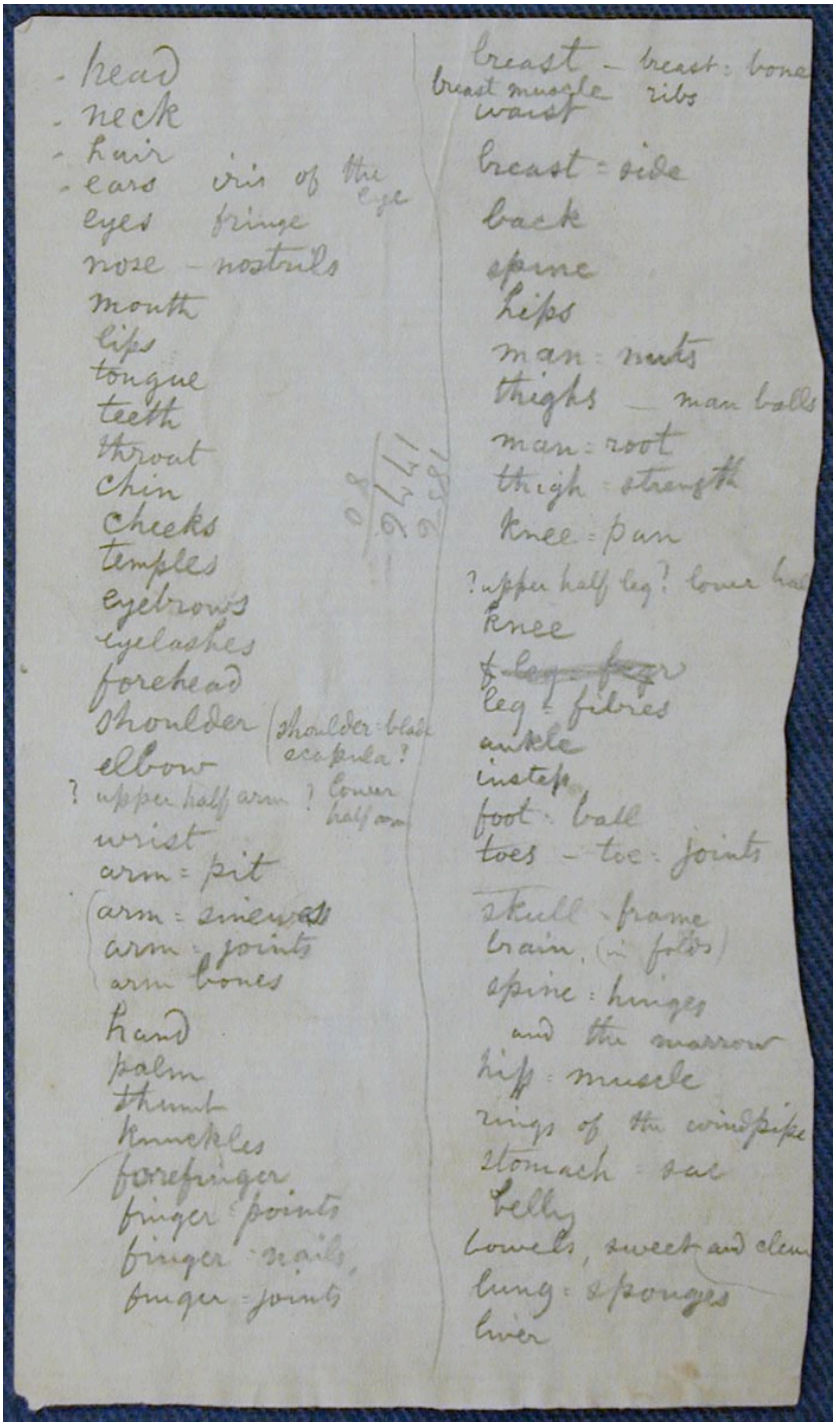

Figure 3: A two-columned list of body parts, from Duke University's Trent Collection, that would eventually be incorporated into "I Sing The Body Electric." Between the two columns, Whitman calculates the number of years between the time of his writing and the year of American independence from Great Britain.[52]

[52] "MS 13." *Trent Collection*. Duke University.

The link between the old man of "I Sing The Body Electric" and the American Revolution suggests that when Whitman is writing about slavery on the eve of the Civil War in the mid-19th century, he is thinking about a comparable time in American history when the British were, in a sense, imposing their own form of servitude. This comparison begins to add the kind of complexity to "I Sing The Body Electric" that a reading of the poem as "egalitarian democracy" neglects. Although the United States would eventually pursue a path of democracy after its separation from the British, the Revolutionary War was not a war for democracy and equality, but instead it was a war for independence and self-government. If Whitman is considering the revolutionary war as he considers the position of black slaves, the parallel becomes less of a parallel about equality and more of a parallel about questions of lawful control and governance.

Although Whitman does not explicitly make an argument linking the rule of the British over the American colonists to the rule of white citizens over black slaves, the way in which Whitman contextualizes the old man encourages a reading that prioritizes his historical memory as the foundation on which more complex interpretation of the poem can begin. Within "I Sing The Body Electric," Whitman introduces dozens of people ranging from mothers and house-keepers to wrestlers and firemen.[53] Each kind of person is introduced in a few words and then the poet continues to the next. However,

[53] Lines 18-28.

to the man "over eighty years old," Whitman devotes ten lines.[54] Not only does he use significantly more poetic space to present the old man, he describes him as "the most beautiful and vigorous of the gang," and, "You would wish long and long to be with him – you would wish to sit by him in the boat, that you and he might touch each other."[55]

Whitman's emphasizing of the old man suggests the man's importance within the poem's meaning structures, and the marginal calculation found in Whitman's manuscript helps uncover a possible reason as to why he is the "most beautiful and vigorous of the gang." The man over eighty years of age, having lived during a time in American history when Americans were not free due to the rule of the British Empire, would be the only person capable of fully appreciating the difference between servitude and freedom because he, unlike anyone else in the poem, would personally remember the pre-Revolutionary War past.

Remembering the past can be understood, for Whitman, as the foundation on which people can make decisions about the present. Although Whitman is a poet, his suggestion should not be minimized for its poeticism unless we would be willing to ascribe the same poeticism to the core business of a company like Google. Google operates its entire search network on a nearly identical principle. Google's search algorithms prioritize billions of historical records of what people have searched for and found useful in order

[54] By the final version of the poem, Whitman gives the old man two additional lines (12 lines total) and makes the entire discussion of the old man its own section.
[55] Lines 38-39.

to make a decision about what kind of information a current user should be connected to.[56]

At its inception, Google assumed it could successfully extract information from vast digital archives by being a tool designed specifically to filter those archives. However, in recent years, Google has learned that accessing past information is not a perfect solution for bridging the gap between humans and knowledge. As a result, Google began correcting its search algorithms by adding logic that accounts for social connectivity. Google searches are no longer solely based on filtering the archives and the ways, historically, those archives have been linked and filtered. Now, Google searches are performed while considering information about the searcher and the people to whom that searcher is connected. Although the temptation may be to decry this kind of knowledge about a user and her social connections as a sort of Orwellian "Big Brother" evolution in technology, Walt Whitman was advocating a similar idea a century and a half earlier. His version appears near the very beginning of "I Sing The Body Electric," when, in the third and fourth lines of the poem, Whitman asks:

> Was it doubted if those who corrupt their own live
>
> bodies conceal themselves?
>
> And if those who defile the living are as bad as
>
> they who defile the dead?

[56] Google's proprietary (and secret) "PageRank" algorithm is used by its search engine to validate the relative content of each website to determine the order for returning search results. Google regularly updates this algorithm, and, over the years, those updates have increasingly focused on accounting for a user's search history.

In this excerpt, the concept of defiling the dead – a person forgetting the past – is presented as an appalling action. As importantly, the corrupting of live bodies – the corrupting of people – is just a bad. These warnings, offered at the very outset of Whitman's poem, introduce the poet's concept of the "body electric." The "body electric," seen from this perspective of historical memory, is a representation of human bodies (i.e. people) as conduits and transmitters of information. The result of this power of transmission inherent in the human body is explained by Whitman when he famously writes that:

> The expression of a well-made man appears not
>
> only in his face,
>
> It is in his limbs and joints also, it is curiously in
>
> the joints of his hips and wrists,
>
> It is in his walk, the carriage of his neck, the flex
>
> of his waist and knees -- dress does not
>
> hide him,
>
> The strong, sweet, supply quality he has, strikes
>
> through the cotton and flannel,
>
> To see him pass conveys as much as the best
>
> poem, perhaps more.[57]

According to Whitman's poem, people naturally convey more information than even the best poems. The result of this power of information-transmission inherent in people is the reason why "you would wish long and

[57] Lines 9-13

long to be with" the man over eighty years of age. By being with the old man, you – Whitman's reader – are connected to the old man's sense of history and, as a result, you too are able to both appreciate your freedom and recognize the complexity of a political system in which not all people are free.

The "body electric" is a poetic assertion by Whitman, echoed (likely unknowingly) a century-and-a-half later by companies like Google, that some of the most successful conveyors of shared cultural knowledge are people, and connecting people with one-another is a means of bridging the gap between humans and knowledge that has been this chapter's recurrent focus. Although a poetic assertion may not normally be the same as either arguable claim or provable fact, what makes Whitman's body electric concept valuable as an interpretive lens for understanding archival extraction technologies is that the same concept can help explain the ascension of person-to-person connective websites (Web 2.0) and their juxtaposition with websites more traditionally structured to connect people directly with information. Is the body electric's power to convey information by connecting people echoed by the digital age's realization that having access to vast repositories of information is of limited use since that information cannot be adequately filtered? Does digitally connecting with people serve the same cultural purpose as extracting information directly from archives? Is digital connectivity an alternate way of extracting information from the vastness of human knowledge and using that extracted knowledge as a resource for making decisions?

These are the kinds of questions that will undergird the following chapters; however, they will do so by separating from archival data extraction

the implication that digitalness is somehow more capable of extracting archival data than, for example, the act of reading a poem, watching a play, or even, as Whitman suggests, seeing a man pass. For Whitman, seeing a man pass is, like reading a poem, a moment of hyperconnection capable of conveying vast quantities of information. But the problem of making full use of the information contained within and transmitted through the man who passes might be the same problem Borges warns of in "The Library of Babel," and it might be the same problem manifested in the digital age as companies try to create better ways to find information within incomprehensibly large repositories of stored knowledge. The problem might be having too much information – whether it be too many men passing by, too many books to read, or too many websites to browse – and not enough time to process everything.

A response to this problem of hyper-archive in the digital age is Web 2.0 – websites that shrug the need to help people directly access archival information in favor of connecting people directly with other people and, by doing so, allow interpersonal connectivity to enable data transfer. But having too much information to filter is not the exclusive domain of the digital age. Prior to a world of Facebook, Twitter, and YouTube, people could have also found value in turning their information technologies into resources for connecting people with other people in order to help mediate the archival chaos associated with vast cultural efforts of social information storage and dissemination.

Recognizing how those historical person-to-person connective technologies can help bridge the gap between humans and knowledge is made easier thanks, in part, to Trygve Reenskaug's MVC metaphor. As the dominant structural metaphor for Web 2.0 connective technologies, MVC provides a blueprint that can be applied to other connective, non-digital technologies that came before. Take, for example, the organizational pattern underlying Walt Whitman's poetry. In a web application, the "model" is the tool for interfacing with the information – knowledge – stored within the application's database. In the case of Whitman's poetry, the "model" is the poet because he is interfacing with the knowledge stored within his audience's shared cultural archive. A website's "view" provides an entry point for the users to interact with and manipulate the underlying model, just as a poem's text provides an entry point for interacting with and manipulating an author. And the "controller," for a Web application, is the component of the digital tool that enables the software to interact with the system in which it is operating. Its textual parallel is perhaps the most instructive part of the metaphor and the one to which I will give the most attention in my next chapters. In literary technology, the "controller" is the combination of critical and social apparatus that contextualize texts within the moments of their reading.

Chapter 2: Developing the Homeric Question

The story of Linus Torvalds adheres to the archetype of what Joseph Campbell, in *The Hero With A Thousand Faces*, calls the "monomyth."[58] It begins as the typical story of a young boy, a "toothy, pale-skinned kid with a blond cowlick living in a suburb of Helsinki, where the weather is cold year-round, save for a few 70-degree weeks in the summer."[59] At the age of 11, Linus receives what Campell's terminology describes as his "Call to Adventure." Linus's grandfather presents him with a "supernatural" device that harnesses the power of energy to perform nearly any task he can imagine. Linus embraces his newfound power. But he quickly discovers a "Road of Trials" on which other people, endowed with the same powers, use their abilities for personal gains. Linus is tempted to do the same, but he refuses to succumb to the vices of others, choosing instead to leverage his growing skills and powers for the good of the world. Torvalds creates something incredibly powerful – more powerful than even the gift he had been given – then gives it to anyone else in the world who wants it, for free, and in doing so he forever changes the existence of mankind. During this journey from young boy to leader of an entire culture, Torvalds cultivates a devoted following of supporters and has the opportunity to amass significant power. However,

---

[58] Campbell's monomyth concept is rooted in the idea that many famous narratives across many generations and cultures follow the same basic pattern. Campbell explains that: "The standard path of the mythological adventure of the hero is a magnification of the formula represented in the rites of passage: *separation – initiation – return:* which might be named the nuclear unit of the monomyth. *A hero ventures forth from the world of common day into a region of supernatural wonder: fabulous forces are there encountered and a decisive victory is won: the hero comes back from this mysterious adventure with the power to bestow boons on his fellow man*" (30).

[59] Learmonth, Michael. "Giving It All Away." *Metroactive*. 8 May 1997. Web. 9 Feb. 2013. http://www.metroactive.com/papers/metro/05.08.97/cover/linus-9719.html

instead of pursuing goals of personal fame and fortune, he crosses "the Return threshold" to pursue a life of relative normalcy and simplicity. In doing so, Torvalds becomes "Master of the Two Worlds," being, in one of those worlds, a husband and father of three children, and, in the other world, a cultural hero of near-mythic proportions.[60]

The story of Linus Torvalds parallels, in many ways, the stories of some of the world's most famous mythic heroes and cultural icons, from Prometheus, to Odysseus, to Moses, Jesus, and Gautama Buddha. Even his name, with its Latinate suffixes and prefixes, helps conjure the image of a tall, muscular warrior battling a world too complex, too powerful, and too corrupt to be successfully navigated by the rest of us “mere mortals.” But Linus Torvalds is not a mythic hero. Not to everyone, anyway. Instead, Linus Torvalds is a software engineer.

The narrative parallels between the life of Linus Torvalds and the mythic “hero with a thousand faces” foreshadow historical parallels about the ways in which Torvalds and his work are being appropriated by his followers and devotees. Analyzing these parallels in a manner similar to the way Joseph Campbell’s work benefits from analyzing similar kinds of historical parallels helps explain one of our most prominent cultural myths: the story of Homer. But before examining those parallels, some work must be done to understand exactly why the work of Linus Torvalds is already beginning to achieve mythic

[60] The stages of Campbell’s monomyth concept present in the story of Linus Torvalds include: “The Call to Adventure” (49-57); “Supernatural Aid” (69-76); “The Crossing of the First Threshold.” (77-89); “The Road of Trials” (97-108); “Woman as the Temptress” (120-125); “The Ultimate Boon” (172-192); “The Crossing of the Return Threshold" (217-228); and “Master of the Two Worlds.” (229-237).

status. Despite the fact that billions of people around the world interact with Torvalds's work every day, few people realize it, and even fewer people understand it.

The importance of Torvalds's work is best contextualized by describing common software usage practices at the time he released his most famous program. While much of today's corporate and university-level digital infrastructure operates on the same basic kinds of software systems as those found on private, personal computers, in 1991, when Linus Torvalds was still a student at the University of Helsinki, personal computers and business computers relied on vastly different operating systems. Most popular among the professional-grade systems was a program developed by the US Department of Defense, IBM, AT&T, UC-Berkeley, and MIT called UNIX. Torvalds, a young computer science student, enjoyed using the powerful UNIX operating system on university workstations, but he hated not being able to access the same kind of functionality on his personal computer. Just to license the software from IBM, Torvalds would have to spend thousands of dollars, and then he would need to spend thousands more purchasing a system capable of running it. Without the resources for personal access to a UNIX system, Torvalds began developing his own software capable of providing UNIX-like functionality on a personal computer.

Creating a powerful operating system for personal computers was not Torvalds's most significant technological contribution. By 1991, emerging corporate giant Microsoft already had the third major iteration of its soon-to-be ubiquitous Windows operating system installed on millions of personal

computers around the world, meaning most PC users already had access to powerful (if not quite as flexible as UNIX) operating systems. What made Torvalds's system so unique was that he made it publicly available online for anyone to download, use, and expand on. Most importantly, he gave it away for free. When he posted the first version of his free software online, the person who managed the remote server where others could download it refused to use the filename Linus wanted.[61] Instead, the server manager posted Torvalds's new operating system under the filename "Linux." The name stuck.

Today, the Linux kernel, the core functionality of Linux-based operating systems, is the underlying software for millions of the world's most commonly used electronic devices including personal computers, smartphones, and GPS navigation systems.[62] But perhaps most importantly for the everyday computer and Internet user, a majority of the world's server architecture – the countless machines linked together by the Internet to store and deliver the World Wide Web – is built on some variation of Linux.[63] In other words, the free software package initially developed and distributed by Linus Torvalds while a college computer science student has become the

[61] Torvalds originally wanted his operating system to be called "Freax" as a play on "free UNIX" and hacker "freaks," a then-popular label for the programming subculture. (Learmonth)
[62] Most commercially available personal computers rely on either the Windows or Mac operating systems (with the Mac operating system being a UNIX variant). However, personal computer distributions of Linux, such as RedHat and Ubuntu, are also popular, particularly in software development professions. On cell phones, the popular Android operating system is built on the Linux kernel.
[63] The reason Linux is so popular is because it is the operating system used for the most common web server environment: the LAMP stack. LAMP is a combination of free and open source software working together to create a general purpose web server environment. The "L" in LAMP stands for Linux. The other softwares are Apache (the web server software), MySQL (a relational database), and either PHP, Python, or Perl serves as the programming language.

foundation for the operating system supporting much of the World Wide Web. Surely such an accomplishment, first engineered by the work of one man, is on the scale of heroic, epic, and even mythic achievement.

While a resemblance to mythic status is readily apparent in the scale and scope of Linux, the operating system also resembles mythic status because of its both its history and its processes of production and replication. Linux is not built or managed by one person; it undergoes constant development from thousands of software developers. Even though Linus Torvalds is recognized as the creator of the Linux kernel, and his name is nearly synonymous with Linux operating systems, Torvalds does not own Linux, he did not create every aspect of Linux, and he does not control how people use Linux. Instead, Torvalds created the original Linux kernel and then shared it with the world.[64] The versions of Linux operating systems commonly used today are not the Linux created by Linus Torvalds; they are versions of Linux operating systems that have been created by the cultures that embraced his core technology. Those Linux derivations – the culturally created Linuxes misleadingly associated with the name of one man – bear a remarkable similarity to other famous cultural myths associated with the name of Homer.

---

[64] Although Linux-based operating systems are commonly associated with the "Linux" name, Torvalds did not create an entire operating system. He created a kernel, which is a low-level abstraction layer within a computer's software architecture that connects an operating system's applications with the underlying hardware those applications are instructing. The name "Linux," while commonly associated with an entire operating system, is misleading shorthand nomenclature, and many Linux-based operating system distributions are co-named, such as GNU/Linux and RedHat-Linux. The naming controversies surrounding Linux-based packages, such as the controversy surrounding Richard Stallman's GNU-project, which preceded Linux and contains many software packages that enable some Linux-based operating systems, are complex, long-running, and, themselves, another contributor to the divergence of the Linux myth and the Linux/Linus name.

"Who was Homer?" is one of history's famously unanswerable questions. It is so prominent in some critical circles that the question itself has a name: the "Homeric Question." The question has troubled so many scholars for so many years that even attribution of its original asking – or at least its concept – is a contentious topic. Some critics attribute the modern understanding of the Homeric Question to Nietzche and his May 1869 lecture at the University of Basel titled "Homer and Classical Philology." Other critics link the question to "eighteenth century British poets and intellectuals [who] began to remember – that is, to create – a Homer that we moderns recognize. Or perhaps it is more precise to say that eighteenth-century British literati developed a Homeric problematic that in many ways remains our own."[65] And the most popular attribution of the concept seems to indicate that the Homeric Question was, in broad terms, first introduced by German philologist Friedrich August Wolf in his well-known and often-cited *Prolegomena ad Homerum*.[66] However, even that assertion is challenged by some Homeric scholars. For example, James I. Porter writes that the Italian political philosopher and historian "Giambattista Vico first articulated the view, well before Nietzsche, that Homer was not a person but an idea (*un' idea*) created

[65] Maureen N. McLane and Laura M. Slatkin make this claim in their article "British Romantic Homer." A few paragraphs later, they attribute the question more specifically when they write, "In eighteenth-century Britain, what we now call the 'Homeric question' (who was Homer? who authored the *Iliad* and the *Odyssey*? when? how?) emerged as a problematic co-forming with several others" (688-689).

[66] Robert Fowler explains that the, "Modern debate began with Wolf, in 1795 who argued that the poems as we have them were put together by a compiler. The result was that it became popular to play a guessing game of what was original to homer and what was the work of someone else" (220).

by the Greeks (though believed in by them), in his *Scienza Nuova Seconda*"[67] which was published in 1730.

The uncertainty surrounding even the original introduction of scholarly questions about Homer underscores the overall complexity of what is meant when referring to "Homer." Was Homer a person? Was he an idea? Was he a complex array of networked poets and stories? These questions have no definitive answers, nor do most critics try to answer them with any certainty because, as Porter explains, "if there is any value at all to 'Homer,' it lies in the very indeterminateness of his definition, in his insolubility, which has provoked intense reflection and so too has served as an instrument of endless debate, contest, and redefinition."[68] For Porter, Homer's value comes not from the success of defining him (or it), but instead from the attempt. The work done to theorize and conceptualize Homer is, itself, the value of Homer. As such, "'Homer' has been," as Porter points out, "good to think with. Or at least, something to think with. Not Homer, but the very idea of Homer."[69]

In the spirit of Porter's suggestion, I wish to "think" with Homer. In doing so, I think I can demonstrate that the "indeterminateness" and inability to define who or what was Homer complexifies critical efforts. The "Homeric problematic" of being able to know "Who was Homer?" is not entirely a problem of being able to know Homer; it is also a problem of trying to

---

[67] Porter, James I. "Homer: The Very Idea." *Arion: A Journal of Humanities and the Classices*, Third Series, Vol. 10, No. 2 (Fall, 2002), 64.
[68] Porter, James. "The Very Idea" (58)
[69] Ibid. Pg. 59. Porter makes another reference to this concept in an article from *The Cambridge Companion to Homer* (Fowler), in which he writes: "Homer, we can safely say, has been 'good to think with', from the earliest appearances of his name right through to the culture and scholarship of the recent past" ("Homer: The History of an Idea", 325).

stabilize a concept of Homer when Homer's output never existed as a stable object. It is, to foreshadow my ultimate parallel, like trying to understand "Who was Linus Torvalds?" if the only evidence of him is Linux. The problem with such a pursuit begins with the first release of Linux. Torvalds did not publish Linux as a complete project. He published it specifically expecting other people to contribute and improve it. For example, in an online Minix[70] newsgroup where he was trying to find an audience for his new Linux kernel, Torvalds posted the following call to action:

> Do you pine for the nice days of Minix 1.1, when men were men and wrote their own device drivers? Are you without a project and just dying to cut your teeth on an OS you can try to modify for your needs? Are you finding it frustrating when everything works in Minix? No more all-nighters to get a nifty program working? Then this post might be just for you.[71]

Torvalds was intentionally publishing an incomplete software that developers could use for their own purposes. As a result, after Torvalds's initial release of Linux, other people began working with it, adding to it, fixing it, removing some parts of it, updating other parts of it, connecting it to other applications, and, overall, embracing the software and making it not Linus Torvalds's software, but a software belonging to the entire Linux community. Working from the current version of Linux, we can never know "Who was Linus

[70] Minix was, like Linux, another Unix clone, but it had an academic-only use license that prevented it from gaining widespread use.
[71] Torvalds, Linus. "Free minix-like kernel sources for 386-AT." Fr. comp.os.minix. 5 Oct. 1991. Web. 9 Feb. 2013. https://groups.google.com/forum/?fromgroups=#!topic/comp.os.minix/4995SivOl9o.

Torvalds." The same thing has, for thousands of years, been happening to Homer because, as Sean Alexander Gurd explains, "literary texts do not survive by simply remaining the same; rather they survive by being reiterated, over and over again, on different substances (papyrus, vellum, paper, .pdf) and in different forms."[72]

Although Gurd's statement about the dynamic nature of literary texts is made in conjunction with a different problematic text from antiquity – Eurpides's *Iphigenia at Aulis* – both his comment and his approach provide a useful foundation for understanding the Homeric Question. Gurd approaches *Iphigenia at Aulis* through the lens of what he calls a "radical philology." Gurd's radical philology attempts to take into consideration the instability of all classical texts. He explains that:

> Critical texts – nearly all classical texts, for that matter, with the possible exception of some inscribed on stone – are produced in a long process of copying, correcting, and publication that affects their form (through corruption, interpolation, conjecture, and emendation, for example). The production of classical texts, in other words, is always located historically, in times and places usually far from their "original" situations. However, the literary scholar nearly always studies texts in the context of their "original" sites of production and reception, with relative

[72] Gurd, Sean Alexander. *Iphigenias At Aulis: Textual Multiplicity, Radical Philology*. Cornell, NY: Cornell University Press, 2005. 44.

> disregard for the long tradition that ends in the critical edition he or she cites.[73]

In order to overcome this problem of historical influence on texts, Gurd uses his concept of a radical philology to approach, understand, and be critical of *Iphigenia at Aulis* not as a single text but as a plural text that incorporates the temporal iterations. In his words, "this radical philology will ultimately be concerned not with 'the' *Iphigenia at Aulis*, located in a specific time and place (late-fifth-century Athens, for example), but with *Iphigenias at Aulis*, essentially plural in form and spatio-temporal position."[74]

Gurd's plural-textual, radical-philological approach is a good starting point for entering a classical text like Homer because he reminds readers that even the critical editions of texts to which scholars often return are highly unstable objects. But, in some regards, Gurd's argument continues to make the same assumption that may be misinforming the Homeric Question. It continues to base its approach on an idea that classical texts – and all texts – have either a stable original, or, at the least, an attempt at a stable original, that, though never recoverable, only became unstable once they were released to the world. But what if Homer never intended a stable text? What if Homer, whoever or whatever he was, like Linus Torvalds, released the framework and idea of a text with the expectation that others would build on his incomplete foundation?

---

[73] Ibid. 23.
[74] Ibid. 21. Hence the title of his book, *Iphagenia**s** at Aulis*, as opposed to the play's singular "Iphagenia."

If such an idea seems radical, consider the revelations about oral traditions and Homer initiated by the work of Milman Parry. Parry's arguments about Homer's oral poetics establish a theory that his poems were composed with the help of pre-defined syntactic units, enabling Homer, or any Homeric bard, to piece together stories by means of a patterned discourse. This understanding of Homer has cultivated the argument that Homer was not a person or a single author. Instead, Parry's work encourages an understanding of Homer as an oral tradition used by ancient Greek bards to disseminate a shared cultural history, or, what John Miles Foley calls "a legend – a way to anthropomorphize the ancient Greek epic tradition." In Foley's opinion, "If scholars have been unable to establish a standard biography and trace the *Iliad* and *Odyssey* to a flesh-and-blood individual, it is, we can conclude, because he simply never existed as such. 'Homer' names the epic tradition as an ongoing whole."[75] For Foley, and others who take a "Homer as Greek cultural tradition" approach to understanding Homeric poetry and oral tradition, Homer is a legend to which other bards appended themselves, like Parry's Guslars did to Cor Huo, Isak, and Hasan Coso,[76] in order to establish their authority and their position within a larger cultural tradition.

Foley's use of the term "legend," as well as any framing of a Parry-esque oral poetic tradition of Homer that confines its use in antiquity, creates a duality of 'now' versus 'then' that implies the modern, contemporary, or

[75] Foley, John Miles. "'Reading' Homer through Oral tradition." *College Literature*. Vol. 34, No. 2, Reading Homer in the 21st Century (Spring, 2007). 7.
[76] Guslars are the 20th century South Slavic oral poets studied by Parry and his assistant, Albert Lord.

otherwise more sophisticated current reader is somehow immune to the possibility of using Homer as a connective information technology through which authority and credibility can be established. The existence of this very chapter and its usage of Homer is all that should be needed to prove otherwise. Homer is, to this very day, and in this document, still being used as a connective information technology.

As I underscore a problematic critical habit of this dualistic 'now' and 'then' approach to reading Homer, Eve Sedgwick's cautions against the trappings of pursuing such bidirectional inquiry in the beginning of *Touching Feeling* offer the reminder that:

> [I]t's far easier to deprecate the confounding, tendentious effects of binary modes of thinking – and to expose their often stultifying perseveration – than it is to articulate or model other structures of thought. Even to invoke nondualism, as plenty of Buddhist sutras point out, is to tumble right into a dualistic trap.[77]

In accordance with Sedgwick's warning, I want to avoid invoking nondualism by, instead, invoking an unabashedly dualistic tool: binary software. Where others see a legend and tradition to which Greek bards attached themselves, I find in Parry's work evidence of a cultural software, not unlike Linux, released some 3,000 years ago and, since that time, constantly updated in order to run properly on the most current cultural technologies. Homeric 'software' need

[77] Sedgwick, Eve Kosofsky. *Touching Feeling: Affect, Pedagogy, Performativity*. Durham, NC: Duke University Press, 2003. Location 73

not be limitedly defined as a tool of antiquity used by Hellenic bards. Instead, Homer seems to be a software used throughout its existence, up to and beyond this very textual moment, to help its users define their present in relation to another's past.

If Parry's work is correct, use of the Homeric software as a technology of information storage and dissemination did not wait until after the Hellenic age to begin.[78] Instead, Parry's work on orality indicates that Homeric poetry was, from its earliest inception, a tool for information storage and dissemination. Without the aid of writing as a primary verbal storage/transmission technology, oral poets leveraged textual clusters to transmit cultural tropes and commonly recognized cultural information blocks.[79] Such textual blocks could range in size from brief (and often remarked on) repetitive noun-epithet formulations like "swift-footed Achilles," to epic-sized structural motifs like the *Odyssey*'s return story.[80] Since, beginning with Parry, and perhaps anticipated as early as the 17th century by Francis Bacon,[81] discussion of Homer's oral syntactic units as

[78] Even if Parry is not correct, the storage and transmission of cultural ideas through stories has a long history I discuss elsewhere in this text, but primarily in Chapter 3.
[79] The Homeric texts hint at this use of verbal clustering by using singular forms of the Greek terms *epos* and *muthos* – usually translated as "word" – when describing a whole speech or story.
[80] These kinds of motifs are discussed often when referencing the Parry tradition, with the specific explanations here being derived from John Miles Foley's article "'Reading' Homer through Oral Tradition," and his book, *Homer's Traditional Art*.
[81] Gerard Passannante points out this interesting anticipation when he notes that: "Bacon's own use of the word *'spargere'* for 'cast' powerfully evokes this generative sense we find in Lucretius, and shifts us again from the language of humanistic despair and skepticism to a language of hope. The very notion of Homer's oral transmission (however vaguely conceived in the early seventeenth century) allows the English philosopher here to figure a dynamic, complex system that in some ways anticipates Milman Parry's thinking many centuries later. By treating Homer's syllables and letters as one unit – by collapsing or eliding the distinction – the philosopher asks

information storage blocks is well documented, I will forgo a more in-depth analysis. I note them here in order to not overlook how, even in its earliest days, the Homeric tradition seems to have been a technology for storage and transmission of information. As importantly, because the Homeric stories were originally cultivated in Hellenic Greece, an argument for the ways in which they connected that culture to its shared sense of history seems frivolous. Of course the Greeks used the *Iliad* and the *Odyssey* and the other stories from the Epic Cycle[82] relating to their cultural heritage in order to connect themselves with that same cultural heritage. It was, after all, their technology. Where the Homeric technology becomes a cross-platform technology is the moment when other cultures begin to adapt it in order to operate it within their own disparate cultural heritages. That cross-cultural tradition of adopting and adapting Homeric technology for preserving a shared cultural heritage gets its most famous example from Rome and Virgil's *Aeneid*, but it actually began well before the founding of the Roman Republic and was a result of the regular contact between ancient Greeks and other Mediterranean cultures.

Despite the clear connection between Rome and Greek cultural histories, the Italian tradition of using Homer as a link to its shared cultural heritage is traceable to the earlier Etruscan civilization. The Etruscans

---

us to rethink the materiality of knowledge in its various forms, suggesting a kind of fluid continuity between media rather than a discreet shift. It is as if printing somehow brings us back to the idea of an oral tradition, to the idea of Homer spreading *from town to town across Greece*, to a vision of social interaction across time and space" (1026-1027).

[82] The Epic Cycle includes the fragmentary remains of other poems, aside from Homer's, related to the Troy Stories. They include details for many of the better known tales (at least in terms of popular culture) such as the Trojan Horse.

appropriated the Homeric legacy before the Romans.[83] They even gave Odysseus, known in Etruscan as Utuse, credit for leading the Etruscans from Lydia to Italy and founding the city of Cortona.[84] Homer was so seamlessly incorporated into Etruscan culture and heritage that archeological evidence from Etruscan burial sites shows clear connections to Homeric poetry, including items ranging from a goblet inscribed with references to the cup of Nestor to tombs decorated with scenes from both the *Iliad* and the *Odyssey*.[85] This use of Homer in burial rituals suggests the Etruscans had adopted the Homeric technology in order to connect themselves to a shared regional cultural heritage. While the specific purposes are likely both complex and largely lost to the natural decay of time, the implications seem clear: Homer was not a proprietary technology closely guarded by the Greeks; he was technology freely distributed to other societies, and the adoption of Homeric technologies by those societies was not only accepted, perhaps it was even encouraged. The same can be said of Linux. It is an open-source technology

[83] What I refer to as the "Homeric Legacy" should not necessarily be limited to the writings attributed to Homer, but is, instead, an intentional simplification of a very complex construction of a culture's shared ancestral heritage. Joseph Farrell makes a similarly useful distinction, so I will appropriate it here. Farrell explains that: "To say 'Greek mythology' is of course not the same as saying 'Homer.' But the historical Greek colonists of the eighth and seventh centuries brought with them stories of a heroic colonization that was the direct result of the Greco-Trojan diaspora set in motion by the fall of Priam's city. The authoritative source to which these stories were traced was naturally Homer, which means not only the *Iliad* and *Odyssey* but also the epic cycle and Homer's followers in other poetic genres.... This dispersion of authority prevents us from making facile assumption about what is and is not 'Homeric.' But even if one takes a conservative approach, adopting a limited purview in order to concentrate on evidence that points to Homer specifically, a conviction emerges that the settling of Italy took place within a Homeric frame of reference" (255).
[84] Farrell, 255.
[85] Ibid. 256-259.

not just freely available for adoption by anyone, but its adoption is actively promoted.[86]

As the Roman civilization embedded the Homeric stories within its culture, and as it incorporated (by culture and force) other civilizations in the region, the Romans made a significant change. In keeping with the metaphor of this discussion, we can view the change to Homer as the Romans releasing a substantial update to the Homeric software – Homer 2.0. Whereas the Etruscans had, for the most part, adopted a Homeric heritage in its entirety in order to trace their cultural heritage, making only minor emendations to the original stories in order to add their founding mythology, the Romans rewrote a critical piece. Instead of tying themselves to both Odysseus and Aeneas, the two founders of Rome according to original regional tradition, the Romans excised Odysseus from the story of its founding and made Aeneas their sole founder. The reason seems to have been to distinguish themselves from Greeks by attaching themselves to a Trojan and therefore historically "other" ancestor. However, despite tracing their lineage to the Trojans instead of the Greeks, the Romans still circumscribed themselves within the same all-encompassing Greek genealogy and cultural mythology.[87]

This substantial emendation of the Greek Homeric software, as opposed to the more simple process of direct borrowing and expansion, denotes a Roman culture that was not necessarily fascinated by Greek culture or an alleged cultural superiority. Instead, as Enrica Sciarrino explains, "just

[86] On the homepage for the Linux Foundation, a non-profit dedicated to supporting Linux, the organization explains that, "The Linux Foundation promotes, protects and advances Linux including supporting the work of Linux creator Linus Torvalds."
[87] Farrell, 255.

as in other traditional societies, so, too in Rome, the 'out there' and the 'back then' were equal foci of energizing spirituality and legitimacy for the living members of society."[88] For Romans, accessing Homerically stored Greek heritage was a way of establishing credibility and authority. The practice was, in ancient Rome, a common occurrence visible in the civilization's earliest luminaries. Quintus Ennius, one of Rome's first highly regarded poets, shares a dream near the beginning of his *Annales* in which he has an encounter with Homer.[89] According to Ennius's dream, Homer's soul, after an interim incarnation in a peacock, was born into Ennius himself. "By presenting himself as Homer reincarnated," Sciarrino explains, "Ennius promotes an understanding of his selfhood as a powerful and yet innocuous conduit for Homer's relocation into Rome."[90] For Ennius, his direct connection with Homer and the Homeric tradition fortifies his historical authority. Homer is, for Ennius, a software of connectivity by which he defines himself and his cultural surroundings in relation to the past – a software Ennius develops, adapts, and modifies in order to provide the foundational cultural operating system in which his text can be authoritatively distributed, read, and understood.

In ancient Rome, the technology of Homer did more than just connect writers with past literary authority. Joseph Farrell explains that "Men like

---

[88] Sciarrino, Enrica. "The Introduction of Epic in Rome: Cultural Thefts and Social Contests." *Arethusa* 39.3 (2006). 459.
[89] Aside from his encounter with Homer, the *Annales* establishes itself through Greek connectivity in other ways, including both style (it is an epic poem written, like Greek epics, in dactylic hexameter) and subject matter (it recounts the history of Rome beginning with the fall of Troy).
[90] Sciarrino, 463.

Cicero, Pliny, Seneca and Lucilius quote Homer casually and sententiously, habitually and deliberately, often amusingly and always with effect, in a manner that suggests not superiority to, but fellowship with their readers and interlocutors."[91] These uses of Homer in daily life reflect a Homeric software being relied on as a communal technology for establishing a sort of shared cultural intimacy. In one famous example, Cicero, writing what might best be described as a recommendation letter to Julius Caesar on behalf of a young man named Precilius, needs to find a way of both acknowledging and overcoming the two men's political animosity. His technology of choice was Homer. Farrell describes this letter, explaining that it is:

> [A] virtual cento of Homeric quotations, leavened by one of Euripedes. By resorting to Homer, Cicero can avoid speaking too directly of his political differences with Caesar and emphasize the humane interests that the two great men of letters share. At the same time, the abundance of quotation is so over-the-top that one cannot read the letter without a smile. This too must have been part of Cicero's intent, to release tensions between himself and his correspondent not only by referring to shared cultural interests, but by doing so in a humorous way.[92]

What Farrell describes as "shared cultural interests" I am defining here as a connective function operating in much the same way people nowadays might invoke a shared interest in a favorite sports team, musical artist, or even a

[91] Farrell, 266.
[92] Ibid. Pg. 264

passion for contributing to open source software projects like Linux in order to establish an interpersonal bond. Homer is functioning as operating software through which Roman society can communicate, and Cicero is accessing a “shared cultural interest” function within that software to remind Caesar how, in essence, "Even though we have had our differences, we are still both descendants of the same cultural greatness that is Roman-ness." As I consider the connection being used, I find myself wondering if either man appreciated that their sense of shared cultural pride was deeply embedded within the heritage of a previous civilization.

In addition to helping Romans establish authority and community, classical Romans used Homeric technology for understanding morality. In a prominent example, Horace uses Homer's ethical principles in his epistle to Lollius Maximus, opening with the following lines:

> Troiani belli scriptorem, Maxime Lolli,
> Dume tu declamas Romae, Praeneste relegi,
> Qui quid sit pulchrum, quid turpe, quid utile, quid non,
> Planius ac melius Chrysippo et Crantore dicit.
>
> (Epist. I.2.I - 4)

> *While you were declaiming at Rome, Lollius Maximus, I was at Praeneste reading the writer of the Trojan War, who says more plainly and better than Chrysippus and Crantor what is fair, what is base, what is useful and what is not.*[93]

[93] Trans. Farrell, 269.

Over the next couple dozen lines, Horace's epistle describes the *Iliad* and *Odyssey* in terms of their moral dimensions, with special emphasis on the self-control to overcome temptation or emotion. The morals Horace is espousing are coming not from Roman custom, but instead from Greek culture brought to Rome by Homeric tradition, demonstrating how the technology of Homer has already, hundreds of years after its creation, preserved and connected a new civilization to the Hellenic sense of morality. Roman morality is not Roman morality, it is Hellenic morality adapted and updated to operate within Roman society. Though the timescale is different, similar evolutions are already apparent in the cultural transferences facilitated by Linux, such as the transference of Unix-like commands and functions to non-Unix devices.

By the time Horace is writing toward the end of the 1st century BC, Rome already has a centuries-old history of using and adapting the Homeric tradition. But a discussion of Rome's connecting through Homer would be incomplete without considering its most famous use and expansion of Homer: Virgil's *Aeneid*. The *Aeneid*, from its original composition through to the present day, is primarily understood as the story of the founding of Rome, and Aeneas himself served Romans as the direct link between themselves and Grecko-Trojan ancestry. The similarities (and differences) of styles and themes between Virgil and Homer are the subject of many books and are too numerous to recount here with any appropriate level of detail or attention,[94]

[94] Some texts include Edan Dekel's *Virgil's Homeric Lens* and Robin Schlunk's *The Homeric Scholia and the Aeneid: a Study of the Influence of Ancient Homeric Literary Criticism on Vergil.*

as is the importance of Virgil and his *Aeneid* in Roman culture from the time of its writing through to the dissolution of the classical state. However, R.D. Williams offers a succinct explanation of the connection that, for the purpose of tracing the development of Homer as a storage and transmission technology, provides sufficient dimension. Williams explains that:

> Aeneas is in a sense a second Odysseus, as he is in a sense a second Hector in the latter part of the *Aeneid*. The Homeric poems had portrayed two great characters of heroic days – the mighty warrior (call him Hector or Achilles; Aeneas is more like Hector, but his situation is that of Achilles); and the man of many devices. Both of them were splendidly efficient in dealing with a hostile heroic environment. Virgil wanted to explore how far their qualities were appropriate for a more advanced civilization, how far the founder of Rome had these same qualities and how far they had to be modified. The heroic world had to be comprehended, embraced, fulfilled by the Romans. The *Aeneid* operates simultaneously on these two time-scales, and this is one of the chief reasons why it is a universal poem. It is an attempted synthesis of heroic and Augustan virtues, and it is particularly in the character of Aeneas that this synthesis is attempted.[95]

In Williams's concept of adapting Homeric qualities for "a more advanced civilization" he is, intentionally or not, understanding civilizations as evolving

[95] Williams, R. D. "Virgil and the 'Odyssey'." *Phoenix*. Vol. 17, No. 4 (Winter, 1963). 273.

technological systems, not unlike computers. Each new version of civilization incorporates distinct technological advances that make it, if not necessarily superior (since superiority is a value judgment), more efficient at doing the work of previous civilizations. As these systems evolve and advance, the cultural software operating within them, like the software operating a computer, needs to be adapted. Virgil's *Aeneid* may be performing this adaptation by taking the sociological tenets of Homer and updating them to operate in the Roman system.

The analogy here between the ways in which civilizations become more advanced and the ways in which computers become more advanced is not coincidental. Nor is it an effort to create a neat metaphor with little analytical value. Instead, the analogy introduces a productive analytical construction. To this point, I have been describing the changes in civilizations as the primary driver of updates to the Homeric operating system. This structure parallels technological advancement in the digital age. As computers evolve, so must the software that runs them. However, in digital evolution, the reverse is also true. While advances in hardware can drive advances in software, more complex software with greater processing needs often drive advances in hardware. Is the same true for literary technologies? Could improvements to the Homeric technology have driven cultural advances?

Possible answers to this question can be seen in the way the Homeric software evolved after Virgil. While Virgil's update of the Homeric software is perhaps the most prominent, drastic, or easily recognized historical update of Homer, it is by no means the only or most recent. By considering the Homeric

tradition not as a stable text but instead as a sort of cultural kernel undergoing constant user development, a Homeric ecosystem evolves. It is an ecosystem, like the Linux software ecosystem, that seems to operate with the purpose of adapting the core software to be able to function on alternate cultural machines. In some cases, it may even be a catalyst for advancement of those machines.

Finding a first prominent example of such adaptations requires little more than looking shortly after, in a historical sense, Virgil's Aeneid to the ways in which Homer facilitates expansion and adoption of early Christian traditions. The multi-deity foundation of the Homeric tradition was incompatible with Christian monotheism. However, Christian leaders would have had trouble making a direct argument that Christianity was better than the paganism of Homeric tradition because the paganism of Homer was older and more established. As Pieter W. Van Der Horst explains:

> 'New' may nowadays imply recommendation, in the ancient world it definitely did not. Aristotle's well known adage... "what is oldest is most venerable"... is an utterance which gives voice to a sentiment that was widespread in antiquity. Modernism or innovation... was, according to the ancients, something reprehensible in many respects, also as far as religion was concerned. The fact that Christianity could not lay claim to ancient credentials – after all, its founder's activities took place only very recently, in the time of Tiberius – militated against the

> dignity, the venerability and the credibility of this new religion.[96]

In order to compensate for the shortcoming of newness described here by Van Der Horst, early Christians leaned heavily on an expansion of Homer that incorporated the Jewish tradition from which they descended. Instead of relying on Jesus to establish their authority, they leveraged Moses and other ancient Hebrew prophets. Thanks to a long Jewish historical tradition established by important historians and chronographers like Demetrius and Josephus, early Christians engineered a version of history that described Homer as being an intellectual descendent of Moses.[97] By doing so, Christians could subsequently argue that the Homeric tradition had been influenced by the Hebrew scriptures, and since the Hebrew scriptures announced the coming of Christ, the ancients, they could reasonably argue, would have both accepted and welcomed Jesus as the messiah and, had they the opportunity, they, too, would have been Christian (i.e. "Followers of Christ").[98] In this way, development of the Homeric software seems to have propelled an advancement of Christian civilization.

These early Christian claims about a Hebraic foundation for Greco-Roman culture may have been as tenuous then as they appear now, but, unfortunately, much of the information about Greek and Roman resistance to Christianity has been lost. As is usually the case in such matters, the victors

---

[96] Van Der Horst, Pieter W. "Plato's Fear as a Topic in Early Christian Apologetics." *Journal of Early Christian Studies* 6.1 (1998). 1.

[97] Homer was not the only important classical Greek figure to be contextualized in such a way. Among others, these manipulations also included Socrates, Plato, and Aristotle.

[98] The introduction to Van Der Horst's article on Plato cited above discusses this historical development in more detail (1-4).

destroyed most of the evidence that would suggest anything but the willing and gracious acceptance of Christian cultural values.[99] However, more important than having the exact details of the transition, at least from the perspective of understanding how the Homeric software may have helped launch a new cultural operating system, is the plethora of remaining evidence that reveals how early Christians launched a new version of Homer – Homer 3.0.

By the time Roman Emperor Constantine I made Christianity the official religion of the Roman Empire in the beginning of the fourth century, the Homeric sense of heroism, religion, and morality had been ingrained in the region for roughly 1,000 years. Early Christians could not simply erase Homer. Instead, they made him a cultural "other" to which Christianity and Christians were vastly superior. The process can be likened to, in modern terms, a negative PR campaign, and the effects were, in a cultural sense, devastating. The early Christian update to the Homeric operating system was to make Homer a barbarian and the symbol of a barbaric culture.[100] He was juxtaposed against Virgil who, as Robin Sowerby explains, was "the great mediating bridge between ancient, medieval and Renaissance and, particularly through the Messianic fourth eclogue, between pagan and Christian culture." In contrast, Sowerby writes that:

---

[99] E. G. Vallianatos provides a detailed, if somewhat embittered, explanation of this process of expanding Christian influences in the region and its consequences in the article "Christians and the Classics: War against Reason."

[100] This barbarizing of Homer seems similar to what companies like Apple and Google have done to Microsoft. Microsoft, once the epicenter of trendy technology, is now widely viewed as a corporate behemoth surviving on old concepts and out-of-date technology paradigms.

> Homer was to prove a point at which the synthesis, always precarious, was liable to break down. Even the more liberal fathers of the early Greek church like Clement of Alexandria and Basil the Great found his gods too much to bear, and in this they were echoing the more famous rejection of Homer by Plato, who had excluded him from his ideal *Republic* (III, 389 and X, 606e). Homer, an archaic poet from a pre-idealist world, remained irredeemably pagan. Of course, the *Odyssey* with its wise hero guided by Athene could yield to allegorical interpretation, but the *Iliad*, always theoretically the preferred choice because of the more heroic context of the Trojan War, intractably resisted allegorical interpretation in its fable as a whole.[101]

In one example of the church struggling with Homeric morality, Clement of Alexandria, an important early Christian theologian, Church Father, and Christian convert writes, "Cease your song, Homer, It's not beautiful anymore. It teaches adultery. We refuse to even listen to fornication."[102]

The barbarizing of Homer by early Christian leaders further underscores how the Homeric legacy seems to be functioning as a technology of cultural information storage and dissemination. Before the dominance of Christianity, Homer, as a technology, reached the point of what digital technophiles call “lock-in.” Lock-in, in technology terms, indicates a

---

[101] Sowerby, Robin. "Early Humanist Failure with Homer (I)." *International Jounral of the Classical Tradition*. Vol. 4, No. 1 (Summer, 1997). 44.
[102] Clement of Alexandria*, Exhortation to the Hellenes* 4.52, trans, G. W. Buttersworth, Loeb Classical Library (1999), 136.

technology that may not be, from a functional perspective, an optimal product, but it remains dominant for a variety of reasons including familiarity, the abundance of other technologies that rely on it, and the containing of massive amounts of data that can be difficult to extract and re-organize.[103] Overcoming technological lock-in requires more than making something better; it often requires making the once-prominent technology reliant on the technology replacing it.[104] The campaign to replace Homer waged by the Church made the Homeric technology reliant on Christian operating technology by, most notably, turning Homer into a corroborating ally of Christian ideologies. Most notably, the Church turned the *Odyssey* into a text primarily recognized as a Christian allegory, and even Homer himself was being depicted as a sort of early Christian saint, prophet, and scholar. In one notable depiction, likely the work of Byzantine scholar Theodore Antiochites sometime around the late 14th century, a hand-drawn portrait of Homer at the beginning of a heavily-annotated copy of the *Iliad* features Homer standing and composing the opening lines of the *Iliad* on a scroll in much the same way artists of the era would occasionally depict notable Christian prophets and evangelists.[105]

---

[103] For a more detailed discussion of Lock-In, particularly in economic terms, see W. Brian Arthur's "Competing Technologies, Increasing Returns, and Lock-In by Historical Events."
[104] A prominent software-based example of this principle was Microsoft's creation of an operating system able to run on multiple PC platforms, which negated IBM's then-significant advantage of having a hardware monopoly. The PC itself became secondary to the software, which, in turn, meant IBM depended on Microsoft instead of Microsoft depending on IBM.
[105] Pontani, Filippomaria. "A Byzantine Portrait of homer"

Figure 4: Ezekiel. Dome mosaic of the church of the Holy Apostles in Saloniki, 1310-14.

Figure 5: St. John of Damascus.Verroia (northern Greece), church of the Anastasis, Dormition fresco (1315)

Figure 6: Homer. Florence, Laur, MS Plut. 32.II, fol. 9. Notice the similarities between this drawing of Homer and the portraits of Ezekiel and St. John.[106]

[106] All images are from Filippomaria's "A Byzantine Portrait of Homer."

Perhaps the acceptance of Homer by Christianity contributed, at least in some ways, to a reignited scholarly interest in Homer at the beginning of the Renaissance. At the time, the original texts of Homer were largely unavailable since few people could read classical Greek. As a result, Homer was recognized primarily as being Virgil's original source, but for little more. The general ignorance of classical Greek impacted much of the work done with Homer by early Italian Humanists. Despite the production of new translations of both the *Iliad* and *Odyssey* by the Italian Humanists, their work was often based on inaccurate understandings and assumptions. In addition, much of their translation work was never completed, with Homer's Italian Renaissance translators often stopping after only a few books. The incompleteness of the work done on Homer by Italian Humanists leads Robin Sowerby to conclude that, "Homer made virtually no impact in the Italian Renaissance and that an appreciation of and taste for Homer were not a part of the rich legacy bequeathed to modern Europe by the early humanists of the Renaissance."[107]

While I agree that much of the evidence supports a failure of the Italian Humanists to properly retrieve Homer, substantial evidence to the contrary suggests that a rich Homeric legacy, as well as a significant update to the Homeric software, was exactly what early humanists of the Italian Renaissance bequeathed to modern Europe. As Philip Ford explains to begin his discussion of Homer in the French Renaissance, "For centuries, Homer's

[107] This particular passage comes from Sowerby's "Early Humanist Failure with Homer (II)" (181). However, the entirety of the two-part article details the many missteps of Italian Humanists in their pursuit of Homer.

poetry was lost to Western Europe, even though the name *Homer* was a byword for the inspired poet, and it was not until Petrarch, the father of Renaissance humanism, turned his attention to Homer that the stage was set for his return."[108] What Sowerby neglects to consider when declaring early humanists as failures in their work with Homer is just how important the attempts of those humanists, led by Petrarch, were to the recovery efforts that reach up to the 21st century and, presumably, beyond. While Petrarch and those who immediately followed him were not perfect in their recovery efforts of an idealized original Homer, they never could have been perfect because there is no original Homer to recover. Instead, Homer was functioning not as static text but, as my metaphor suggests, a sort of cultural software being updated to operate within the then-current cultural moment as well as forcing advancements to the civilizations in which it was being used. If the contribution of Petrarch and other Italian Humanists is viewed as an attempt to restore Homer, then it is most certainly a failure, but if the work of the Italian Humanists was to re-adapt the Homeric technology for a different age, then they were amazingly successful.

By the time Petrarch is first beginning to consider Homer, the culture in which he is operating is no longer concerned with the restoration of ancient Greek Hellenic values. Christianity's near-authoritarian dominance on religion in Western Europe had been in place for the better part of a millennium, and what Petrarch seems to have been experiencing was likely a loosening of the grasp of Christianity that had kept Homer locked away as the

[108] Ford, Philip. "Homer in the French Renaissance." *Renaissance Quarterly* 59.1 (2006). 1.

anti-Christian source from which the Christianized Virgil descended. Sowerby even makes reference to this loosening of cultural (i.e. religious) suspicions about the ancient texts when he compares the concerns of early Christian intellectuals who dared to consider the classical texts with those of Petrarch, writing that:

> Early Christian intellectuals in the west, like Jerome and Augustine, had come to regard the enthusiasm they had felt for the pagan literature in which they had been schooled as worldly vanity that threatened their salvation. Jerome's nightmare was that he would be judged to be more of a Ciceronian than a Christian, "for where your treasure is, there shall your heart be also." Here we may not feel that Petrarch is seriously troubled that he will similarly be dismissed for being a Virgilian; nevertheless there is an unresolved tension between two seemingly irreconcilable viewpoints.[109]

Petrarch no longer needed to be concerned with being labeled "anti-Christian" for appreciating Homer. Instead he seems to feel the natural tension between two cultural operating systems that are, on certain core levels of belief, clearly incompatible. In comparison, much of the current Judeo-Christian world operates largely within the same cultural operating system as Petrarch, and when reading Homer, the same tension can still be felt.[110]

---

[109] Sowerby, "Humanist Failure (I)," 43.
[110] Here again I point to E. G. Vallianatos article as an example of this tension and the people considering it.

The contribution of Petrarch and the other early Italian Humanists to "modern Europe" does not need to be viewed as their restoration of Homer as the dominant cultural operating technology. Instead, we can view them as using the name recognition of the Homeric legacy to turn Homer into a niche resource technology, not dissimilar to what often happens when a highly trafficked website's popularity wanes.[111] Beginning with Petrarch, the Italian Humanists reintroduced Homer as a useful ancillary resource used for romanticized cultural linking and historical lineage. That legacy is clearly visible in the brief adoption of the Homeric tradition by 16th century France. For 16th century France, "The Homeric epics were like an archaeological sieve," Ford explains. And from that sieve, the French would discover:

> [N]ot only information about the societies that they appeared to portray, but also the metaphysical and religious beliefs that underpinned these societies and which were revealed by Homer in the poems. Inspired by a syncretist view of the past, these views reinforced the Pseudo-Plutarchan view of Homer as originator of all the arts and sciences, supreme poet, and unsurpassed genius. The following centuries would see this view of Homer contested, particularly at the end of the seventeenth century and the beginning of the eighteenth when the *Quere... d'Homere* was at its height. But for a brief period in the French

[111] Digg.com, MySpace.com, and Friendster.com were all, at one time, some of the most heavily trafficked websites in the world. Now they exist in alternate forms, serving niche communities. As of the time of this writing, Digg, the former social news aggregator, primarily operates an RSS feed reader, MySpace is a social network for musicians, and Friendster is the platform for a social gaming community.

Renaissance, the legendary Homer was held in the highest possible esteem by France's intellectual and cultural elite.[112]

The likely reason for the brief French adoption of Homer was as much political as it was cultural. The ascension of Homer in France coincided with the reign of François I, a monarch better known for his interest in art and culture than his success in war. As François was being raised, the Italian Renaissance were influencing French culture, and François's education was largely driven by the era's humanist trends. From that education, François may have found in Homer a perfect military ally against his hated rival, Holy Roman Emperor Charles V.

In the early 16th century, during the reign of François I, classical Greek became a standard component of the French education system,[113] legitimizing Greek studies and giving the French nobility a strong background in ancient Greek culture – a culture that could be directly linked with Homer. This monarchical support of Greek would have been, at the time, in direct opposition to the Catholic Church's Latinity, which was, more importantly, the same Catholic Church to which Charles V's power was aligned. In addition, the land encompassing the birthplace of Homer was under the rule of François's occasional ally in the fight against Charles V, the Ottoman Empire.[114] Homer, and classical Greek, had seemingly become a technology

---

[112] Ford, 23.
[113] A popular belief at the time, known as *Celt'-hélénisme*, held that French was not derived from Latin, but instead from Greek (Rothstein, 735).
[114] Rothstein, Marian. "Homer for the Court of François I" *Renaissance Quarterly*. 59.3 (2006). 734.

with which France both connected itself to an alternate tradition than the Church in Rome and connected itself to its military ally.

This political and military realignment was echoed by the era's Homeric translations and representations. François commissioned translations of the *Iliad* so French nobles could access the text as a source of history and noble conduct. The emphasis in these texts was not as much on accuracy as adaptation for the purposes of conveying and reiterating the beliefs of the French nobility. They would emphasize, for example, and not entirely accurately, a Greek monarchical structure with a clearly defined lineation between ruler and subjects as well as the importance of subservience. Woodcuts and other similar depiction mediums would also represent Homer's epic in French military styles by, for example, illustrating tented camps, a practice common to the French army but, according to Homer, different from the Greeks' practice of living on their boats. The overall convergence of textual and artistic themes representing the *Iliad* formed in France what Marian Rothstein describes as a "pro-Greek and pro-royalist viewpoint."[115] That viewpoint operated on Homeric technology and the technology's ability to help François access and distribute a shared cultural heritage around which his people could rally.

The connection to a Homeric lineage enabled by France's use of the Homeric operating system extended far beyond politics and deep, if somewhat temporarily, into the French cultural fabric. In the early 1500s, many French subjects held the common belief that they were direct

[115] Ibid. 763.

descendents of Troy. Although factual belief of France's Trojan lineage abated by the 1540's, French propaganda continued to leverage a figurative sense of Trojan origins throughout the century as a means of laying claim to its artistic and cultural superiority.[116] Hugues Salel, a 16th century French courtier, cleric, and poet who produced a verse translation of the *Iliad*'s first 10 books, makes a reference to this heritage in his prefatory verse epistle when his speaker, Dame Poésie (Lady Poetry), refers to Homer's prophetic powers of recognizing the Franco-Trojan lineage by saying:

> I will abstain for the moment from declaring
> how the gods wanted to grant him
> prophetic powers, by which he predicated
> the authority, the reign, and the respect
> that Trojans, having survived great dangers,
> would one day have in foreign lands.[117]

This sense of Homeric lineage provided French humanists with a different kind of authority than Homer gave François – it gave them moral authority. "The *Iliad* and the *Odyssey* were considered to be the source of all the arts and sciences as well as the philosophical schools," Ford explains, and allegorical readings provided the "moral wisdom contained in the two epics."[118] From this sense of moral wisdom, important French intellectuals in mid-16th-century France, including Guillaume Budé and Jean Dorat, established a Homer who was the pinnacle of Greek cultural achievement.

---

[116] Ibid. 735.
[117] Salel, Hugues. "Dedicatory Epistle to Homer's *Iliad*," II. 145-50. Translation fr. Bernard Weinberg. *Critical Prefaces of the French Renaissance*. 1950. Reprinted New York, 1970.
[118] Ford. 2.

This appropriation of Homeric cultural technology by the French was not as generous to the Greek bard as might at first appear. During this time, the French were actually deploying Homeric cultural technology not specifically to praise Homer's texts and culture, but to instead establish their own culture as Homer's cultural equal, or perhaps even the cultural superior. Again, development of the Homeric software seems to have been advancing the cultural operating system on which it was running. Until this point in their history, the prevailing French assumption was to identify Homer's epics as a single peak in the world's artistic and cultural production. Sixteenth century French intellectuals instead began to define their own texts by those of Homer until they began to perceive their works as containing similar merit.[119] Upon reaching the point of perceived equality, the decline of Homer's authorial status and the popularity of the Homeric operating system in France was inevitable. In the later part of the century, the posthumously published critical writings of Julius Caesar Scaliger altered French Homeric perceptions. Emphasizing the kind of realism that would become a cornerstone of the French classical approach of the coming century, Scaliger describes Homeric characteristics, such as his epithets, as "often cold and puerile or inappropriate to their context. What is the point of saying of Achilles when he is weeping that he is swift-footed?"[120] Scaliger would often compare Homer with Virgil, and, not having the Parry-inspired understanding of Homer's oral devices, saw in Homer irredeemable stylistic failings that made his antiquated

[119] Rothstein. 734.
[120] Trans. fr. Ford, 22.

texts incomparable to those that followed, especially those produced by the French.

Even though 16th Century France quickly went from one of the most eager adopters of Homer to one of the technology's biggest critics, the French adaptation of Homer represents another example of how the Homeric operating system may have again been transformed and been transformative. When the French began to rapidly adapt Homer, they needed a cultural operating system that provided a much-needed military and political ally against Catholicism and the powerful Spaniards. But being a politically separate entity was not useful enough functionality. The French also needed a rich cultural identity. To provide that cultural identity, the Homeric technology appears to have been transformed from one that set the standard for high culture, as it did for the Roman Empire and eventual Holy Roman Empire of the Catholics and King Charles, into one that was inferior to French culture. Homeric technology was thus being deployed first as a measuring stick and then as a standard to be surpassed in every way.

While the French were, in a sense, making Homer obsolete, the English were, in some ways, redeploying Homer, and, in other ways, deploying Homer for the first time. This dichotomy comes from distinguishing Homer from what are known as the "Troy Stories." The Troy Stories are the vulgarized tales of Greek epic that were popular throughout Western Europe during the Middle Ages. They are presumed to be the descendants of the Epic Cycle, and they contain many of the more famous moments of the Trojan epic (Shakespeare's *Troilus and Cressida*, for example, is a product of the Troy

Stories and not Homer), but they are less Homeric epic and more romanticized narrative.[121] Until the end of the 16th century, the Troy Stories were also the dominant vessel through which the English received the Homeric tradition. Howard Clarke describes the Homeric tradition in England and much of Western Europe prior to the 16th century rediscovery of Homer in the following excerpt:

> [F]or the thousand years of its heyday in European literature, the Troy story wavered between the attractions of history and romance. Its authors thought they were preserving a historical record; in fact they were embroidering a magnificent fiction. If on balance the romance element predominated, it is because there was more there to delight than to instruct, and its writers had to supplement their limited historical resources with their unlimited imaginations. Still, the story was immensely popular, in part because of the inherent interest of the subject (the siege of Thebes was its only competition) and Troy's dynastic links with Western Europe, in part, too, because the Crusades offered a contemporary parallel of war in the east, but mainly because of its unique combination of history's authority and romance's fascination. It catered to European nationalism by reminding its readers of their illustrious forbears and yet invoked the mysterious East only then giving way to Crusaders and travelers.

[121] Clarke, Howard. *Homer's Readers: A Historical Introduction to the* Iliad *and the* Odyssey. Newark, DE: University of Delaware Press, 1980. 20.

> Its warriors fought with familiar weapons and yet bore names that were prestigious and exotic. Calchas was a "bysshop" and yet he had a beautiful daughter. Classical in its nomenclature and medieval in its setting, the Troy story was universal in its themes and unmatched in its popularity.[122]

In this atmosphere of romanticized and dramatized tales of distant epic and chivalry, actual Hellenic Homer, primarily available in unreadable Greek texts, heavily invested in a world of non-Christian gods, and encumbered by exhaustive lists, was likely considered – to deploy an anachronism – boring.

This perception of Homer as perhaps "less Homeric" than the more popular Homeric tradition at the time seems to have been, like in early Renaissance Italy, what helped facilitate the gradual re-adaptation of the original Homeric software. Earliest among the notable English authors helping to develop a new version of Homer more appropriate for the society in which it was being used was Spenser. Spenser was composing *The Faerie Queene* in the 1580s and 1590s, immediately in the midst of the publication of the first two partial translations of the *Iliad* into English in 1582 and 1598. In addition, Spenser was noted for his knowledge of classical Greek, and he considered himself the poetic successor of Thomas Drant who, in the 1560s, embarked on the first recorded attempt to translate the *Iliad* into English.[123] Spenser scholars disagree as to the extent Homer influenced Spenser and *The Faerie Queen*, with some arguing that any connection to Homer is through the

---

[122] Ibid. 55.
[123] Demetriou, Tania. "'Essentially Circe': Spenser, Homer, and the Homeric Tradition." *Translation and Literature* Vol. 15, No. 2 (Autumn, 2006). 153.

more popular Troy Stories, while others believe Spenser used a predominantly text-based and classics-based Homeric knowledge.[124] The existence of these arguments likely offers more certainty about the topic than the arguments themselves. Spenser's actual engagement with Homer would have to have been a mixture of both classical knowledge of the Homeric tradition and cultural engagement with the popular romanticized versions because, when Spenser was writing, the operating version of Homer was a popularized and historicized mixture.

Later English authors would continue to develop the Homeric operating system toward the classical and textual traditions that further suggests Homer's functioning as a technology of cultural contrast and comparison. Milton, whose knowledge of classical Greek was nearly flawless, makes a number of direct and indirect references to both the *Iliad* and the *Odyssey* in *Paradise Lost* that seems to, in a way, make Homeric cultural heritage secondary to the primacy of his Judeo-Christian narrative.[125] Alexander Pope, in contrast, worked to popularize Homer. He was, according to Penelope Wilson, "intent both on rescuing Homer from the depredations and irrelevancies of scholars and critics, and establishing the Homeric poems at the centre of a new literary market-place."[126] In so doing, Wilson suggests

---

[124] Douglas Bush is an earlier 20th century example of arguing for the story tradition (*Mythology and the Renaissance Tradition in English Poetry*. 1932; reprint New York, 1957); Gordon Teskey is a more recent example of the argument where he offers an uncertainty connecting Spenser directly with Homer ("Homer," in *The Spenser Encyclopedia*, edited by A. G. Hamilton. Toronto, 1990.); and Demetriou's article is an attempt to bring more certainty to the argument, though her arguments still show a mixed usage that seems to be the most likely scenario.
[125] Wilson, Penelope. "Homer and English Epic" fr. *The Cambridge Companion to Homer*. Ed. Robert Fowler. Cambridge: Cambridge University Press, 2004. 277-280.
[126] Ibid. 280.

that Pope's translations and popularizations of Homer were among the more important active developments to the old epic tradition that would foreshadow Wolf's *Prolegomena* and the 19th century popularizing of the Homeric Question.[127]

What Pope, and others invested in reviving and translating Homer, had attempted to accomplish by this point in the history of what I have been describing, both directly and indirectly, as Homeric software development is to update a romanticized and Christianized version of Homer back to the original Homer. Not that they ever could. As Gerard Passannante explains when describing Francis Bacon's understanding of Homer:

> Nothing persists. Everything is conserved and made new. If by chance all the atoms of a person – or, as it were, letters of a text – came together in precisely the same way, there would still be a difference. In the space of this difference the poet posits the possibility of reception. It was by a similar logic, one could say, that Bacon himself came to resist the claims of fixity and permanence promised by print while continuing to celebrate the dynamic possibility of the medium.[128]

For Bacon, an original Homer could never be restored because the context in which Homer was being experienced could never be restored. Instead, Bacon recognizes and celebrates the "dynamic possibility" of the Homeric technology in a way that explains the nature of the Homeric Question that would soon

[127] Ibid. 285-286.
[128] Passannante, 1036.

follow. With versions of Homer that were no longer deeply embedded in the romanticized Troy Stories, scholars would have been left with some semblance of an "original text" and a long tradition of decoding. The "dynamic possibility" of the "restored text" would have been to continue doing to Homer what they had done for so many years, except without a text to restore, they could begin work restoring the old bard himself.

The restoration of Homer that would lay the foundation for our current implementation of the Homeric operating system took two not-entirely-distinct routes, one textual and the other biographical. The textual restoration of Homer was not, technically, an actual restoring of the Homeric texts to their original form because the focus had become translating Homer from the original Greek into an acceptable English adaptation. In the 1800s, Britain alone produced nearly as many translations of Homer as there are years in the century.[129] This dramatic increase in the volume of translations suggests an intense desire to better understand the "original Homer" by organizing his language in a way that could be properly realized and appreciated by contemporary readers. But the work of translation could not be accomplished without operating in concert with the work of understanding Homer the person and the historical world in which he existed. As a result, the work of translating Homer initiated the development of a concept of "Homer" that has occupied much of the 19th, 20th, and, to this point, the 21st century. Understanding the text of Homer has apparently become, for the modern reader, a task of understanding the personage of Homer.

[129] Porter, "Homer: the History of an Idea," 338.

Wolf's *Prolegomena* is the most easily recognized early example of the increasing emphasis on a biographical focus.[130] Wolf argues that:

> The Homer that we hold in our hands now is not the one who flourished in the mouths of the Greeks of his own day, but one variously altered, interpolated, corrected, and emended from the times of Solon down to those of the Alexandrians. Learned and clever men have long felt their way to this conclusion by using various scattered bits of evidence; but now the voices of all periods joined together bear witness, and history speaks.[131]

With this argument, Wolf turns the Homeric texts into what Porter describes as "an archaeological site, with layers of history built into them in a palpable stratigraphy: the disparate effects of multiple compositional layers… and the intrusive hands of editors could all be felt in the poems."[132] From within that archaeological site, Homeric scholarship unearthed and continues to unearth a cornucopia of perceived 'evidence' about the true Homer and his texts, which is often as likely to disagree with the artifacts produced by others as it is to agree. Nietzsche's "Homer and Classical Philology" lecture distantly describe the varying opinions, though in doing so he proffers his own highly contentious suggestion that "The name of Homer, from the very beginning, has no connection either with the conception of aesthetic perfection or yet

---

[130] Though, as Robert Fowler notes, Wolf's work was prefaced and, in part, initiated in 1788 by Jean-Babtiste Gaspard d'Ansse de Villoison's publication of a study on a 10th century *Iliad* manuscript containing substantial ancient scholia and marginal annotations suggesting corrupt versus.

[131] Wolf, F. A. *Prolegomena To Homer*. Trans. Anthony Grafton, Glenn W. Most, and James E. G. Zetzel. Princeton, NJ: Princton University Press, 1985. 209.

[132] Porter, "Homer: the History of an Idea," 337.

with the *Iliad* and the *Odyssey*. Homer as the composer of the *Iliad* and the *Odyssey* is not a historical tradition, but an *aesthetic judgment*."[133]

What Nietzche labels an "aesthetic judgment" is, in many ways, what I am attempting to define as a technology of information storage and dissemination. In contrast to my explanation, Nietzsche's "aesthetic judgment" concept gets undercut by the current preferred iteration of biographical Homeric exploration and argumentation derived from Milman Parry and his early 20th century work on South-Slavic oral traditions. In his research, Parry counters the Nietzschian sense of aesthetic judgment with an assertion that Homer is a sort of programmatic framework.[134] Parry's work presents Homer as a kind of programmatic oral tradition in which narrative is formed by piecing together culturally pre-fabricated syntactic units of meaning.

Parry's widely accepted work on Homer is a telling contemporary example of how even our current scholastic culture appears to be adapting the Homeric operating system to run within the technological iterations of our own cultural systems. Egbert Bakker, for example, in an article titled "Homer, Hypertext, and the Web of Myth," uses Parry's concept of oral, syntactic blocks of meaning as an entry point for an argument about how hypertexts are restoring recorded discourse to a more natural state of communication. For Bakker, hypertexts and the Internet are reversing the course of the artificial

---

[133] Nietzsche. "Homer and Classical Philology" tr. John McFarland Kennedy. Web. 9 Feb. 2013. http://www.gutenberg.org/files/18188/18188-h/18188-h.htm.

[134] I describe Parry's argument as a "programmatic framework" not as a repetition of Parry's words, but because the concept of a programmatic framework accurately describes Parry's explanation of Homer even if Parry had no concept of a software-based programmatic framework. A more detailed consideration of software frameworks is addressed in chapter 3.

linearity imposed on literature that seems to have begun with Oral poets like Homer, crystallized around Aristotle, and appeared to cement itself through technologies like books, printing, and even electronic literature.[135] In the discoveries of Homeric scholars arguing for a concept of oral poetics, Bakker has found meaning in oral poetry's relationship with the World Wide Web. When explaining the contributions of those scholars, Bakker writes:

> Oral poetics [...] stressed parataxis over hypotaxis on the level of sentential syntax, and episodic structure over linearity and hierarchical relationships on the level of plot. The linear order of beginning, middle, and end came to be replaced with images such as beads on a string, a narrative sequence that provided the oral poet numerous points of entrance and exit. In departing from an Aristotelian poetics based on linearity and hierarchical order, the oralists came close to the advocates of hypertext in some important respects.[136]

Regardless of the validity of Bakker's assertion, which is not at issue here, Bakker's decision to view the most recent milestones in Homeric scholarship through a lens of digital-critical inquiry demonstrates how

---

[135] Pamela Jennings, in her article "Narrative Structures for New Media: Towards a New Definition," provides a helpful explanation of Aristotle's influence on the Western concept of narrative as a linear model. She explains that, "The written culture's notion of narrative derives from the theory of dramatic progression expounded by Aristotle in his *Poetics*. The *Poetics* presents a strict guideline for the drama to follow from beginning to end: the narrative increases in intensity to the climax and then gradually reaches an end parallel in tone to its beginning" (346). Expanding on a similar concept, Michelle Glaros, in "<frame>ing Representations of the Web," notes how "Even a concept as simple as the Aristotelian plot structure of beginning, middle, and end constitutes a frame that identifies and legitimates electronic writing as literature while simultaneously, and perhaps unnoticeably, constraining such work" (L 1718).

[136] Bakker, Egbert J. "Homer, Hypertext, and the Web of Myth." Web. 5 Oct. 2011. http://www.philo.umontreal.ca/documents/cahiers/Bakker_homer.pdf.

the Homeric operating system is still being adapted for use in our current culture. The social and cultural adaptability of the Homeric technology allows for people like Bakker to argue that: “The loss of control in the age of the internet, I submit, is mirrored by its reversal in archaic Greece: the gradual establishment of control exercised by the Homeric tradition as a concerted effort to overcome the vagaries of the oral tradition.”[137] Bakker is using the Homeric technology as a tool for understanding the ways in which digital technologies are changing expectations of traditional linear narrative. Surely Homer – whoever or whatever “he” was – could not have anticipated his texts would be used as Bakker uses them, just as Linus Torvalds could not have predicted Linux would be used to power millions of cell phones and servers. Similarly, just as Linux developers who updated the software after its initial release by Torvalds are – literally – enabling my physical act of writing this chapter on a computer, Parry’s development work on the Homeric technology is enabling me to deploy Homeric technology to operate this chapter’s software-like reading of Homeric production.

But Parry's thesis provides more questions than it does answers. Included among those questions is whether or not the "Homer" of tradition was a real person or if "Homer" was itself a sort of pre-fabricated syntactic unit which other epic bards could summon in order to convey a culturally-encapsulated sense of meaning. In other words, despite the discipline-altering insights that Parry's work provides, it only re-adapts the same Homeric

[137] ibid. 13.

Question into a form capable of being processed from within our own cultural operating system. In that re-adaptation exists perhaps the most telling sign that information technologies – including those of a digital slant –have the potential to be tools of literary critical inquiry. As Stephen Ramsay explains, "The goal [of literary criticism] is not to arrive at the truth, as science strives to do. In literary criticism, as in the humanities more generally, the goal has always been to arrive at the question."[138] Analyzing Homer as a technology of information storage and dissemination can do just that – it can lead us to better Homeric Question, which we can now properly derive.

Even from the brief overview provided here – which, to borrow from Nietzsche, "keep[s] all minor details rigorously at a distance"[139] – by necessarily picking and choosing only a few well-defined examples from a textual history as complex as any in the world, I hope to have clearly implied, and will here overtly state, that any attempts to restore an original Homer – whoever or whatever "he" was – are, from the perspective of feasibility, ultimately untenable. But I doubt any critics seriously pursuing the task would argue otherwise. They likely realize that Homer is, as noted near the beginning of this chapter, a tool scholars can think with. However, prioritizing the act of thinking *with* Homer should not obscure an opportunity to think *about* Homer. As elusive as the answer to the Homeric Question may seem, if Homer is indeed a form of connective software being regularly developed to fit within the current cultural hardware using it, then we would at least have

[138] Ramsay, 69.
[139] Nietzsche, "Homer and Classical Philology."

one morsel of certain information: the "Homer" we know and use today is not the "Homer" created by the software's original author; it is the "Homer" created by the cultures that have embraced Homeric technology. As frustrating as this morsel of certainty may be to those wishing to know, with absolutely certainty, "Who was homer?", consider also how the available certainty provides a glint of hope. By being a culturally developed software, the Homeric operating system would have some of the same mechanisms which software development uses to both preserve and restore originals. So instead of beginning with the Homeric Question (i.e. instead of wondering "Who was Homer?") and immediately being overwhelmed by thousands of years of history, I propose first transposing the question to a similar culturally-adapted object with a less-overwhelming history. I propose asking the question: "Who was Linux?"

Having the luxury of (indirectly) witnessing the creation of Linux and watching its cultural development, we can confidently know the question "Who was Linux?" is inherently misleading. Linux is not a person – it is a software developed by Linus Torvalds. The misleading question helpfully evokes the problem I suggested near the beginning of this chapter when I wondered about the difficulty of answering the question "Who was Linus Torvalds?" if the only evidence of him is Linux. Both questions – “Who was Linux?” and “Who was Linus Torvalds?” – force inherently unavailable answers. The first question expects an answer about a person when the subject is an object (i.e. the Linux kernel). The second question expects an answer about a person while the only information about him comes from a

community-developed “text” with few similarities to the author’s original. But knowing that the questions themselves are flawed allows us to break the errant "Who was Linux?" question into three valid questions: Who was the original creator of Linux? What was the original version of Linux? And what can be learned about the original creator of Linux based on what is known about Linux's original version?

The Homeric Question might contain an identical scenario. Asking "Who was Homer?" necessarily implies that "Homer" the inventive entity is the same as "Homer" the cultural software. Instead, the Homeric Question can be split into three questions: Who was the original creator of the Homeric texts? What were the original versions of the Homeric texts? And, based on what is known about the original versions of the Homeric texts, what can be learned about their creator(s)?

These three-question parallels seem to provide a useful strategy for uncovering information about the original author(s) of the Homeric texts while only having derivative versions of those texts. The strategy would be the same one used to uncover information about Linus Torvalds based on having access only to different derivations of Linux. Such an approach offers two significant advantages. The first advantage is primarily conceptual: a comparison of Homer and Linux underscores the inability of ever deriving a true sense of personhood. Just as Linux is clearly a continuously developed software and not the person who first created it, paralleling Homer and Linux suggests that Homer should similarly be understood as a continuously developed text and not the person who first created it. A second advantage of

comparing Linux to Homer is that it would allow us to consider Homeric software development through the lens of the same processes used to ensure successful iterations of new Linux versions.

In the same way the cultural environments in which Homer operates are always evolving, so, too, are the computer environments running Linux. As a result, Linux, like Homer, is under constant development. Also like Homer, the Linux kernel's codebase is an exceedingly large text requiring the attention of numerous developers.[140] Because Linux is so large and complex, the Linux community, like any multi-developer entity managing large software projects, requires a way of organizing multi-user development to ensure software stability. Their resource of choice to manage this task is a distributed version control and source code management (SCM) system. That system does for Linux what thousands of years of critical scholarship have done for Homer: it maintains software operability.

The history of the Linux source code management solution is, in itself, an interesting and complex story that ultimately resulted in Linus Torvalds developing his own SCM, called Git, which, like Linux, he made freely available. By doing so, Git's impact on software development has been almost as impactful as Linux itself. Perhaps, in some respects, Git is to Linux what the *Odyssey* is to the *Iliad*. But a detailed discussion of Git and the version control system history of Linux in this chapter would complicate and distract

[140] The original version of Linux, released by Torvalds in September 1991 as version 0.01, had just over 10,000 lines of code. The most recent version of the Linux kernel, released in 2012, has 14,998,651 lines of code ("Summing up 3.2").

from the primary explanation of why the Linux version control system is relevant to Homer.

SCM systems like Git retain different versions of a piece of software that are each linked to different iterations and stages of the software's development. Those archived versions record the history of all the software's changes. While a software program's users see only the working version, the software developers have a separate version on which they can continue development. In addition, when multiple developers are working on the same software, the SCM helps the developers avoid overwriting the work of other developers in order to prevent important changes from being overwritten and causing the software to stop functioning. To accomplish this separation of public and private code, and to maintain a functional development process, source code management practices for large software projects generally rely on one of two options: either the developers use a centralized source code repository from which everyone pulls a version of the software, modifies it, then returns it to the central repository; or the developers each have a fully working version of the software on their own computers which they can develop and then share their development work with co-developers.

Linux uses the latter of these two SCM options – a distributed source code management solution. As a result, Linux developers are always working from their own private codebases, and the results are rather Homeric. In one respect, developers can contribute to Linux itself, the central software, by submitting (i.e. "publishing") changes and updates to the software in order to improve the way it functions in evolving operating environments in much the

same way Homer scholars (or, since such work is not specific to Homer, any textual scholars) can contribute to the primary Homeric software. Parry did this kind of work when he published his research on Homer's oral heritage. Wolf did it when he published his discussion of Homer as a conceptual tradition. And similar updates to the Homeric software can be traced back through the centuries, to the Renaissance French, Italian Humanists, Byzantines, Romans, Etruscans, and even the ancient Greeks.[141] In a second relevant parallel, the distributed nature of Linux has resulted in the development of different Linux distributions which might best be described as pre-packaged versions of Linux-based operating systems designed for different end-user scenarios, ranging from different desktop operating environments[142] to different hardware environments like mobile phones and web servers. These Linux distributions are, in many respects, translations of the original Linux kernel which enable different hardware environments to read and make use of Linux in much the same way different translations of Homer allow different cultures to use the Homeric software, as well as different versions of that Homeric software according to what translators perceive to be its most important components. In this respect, every translation of Homer, from Stephen Mitchell's modern, commercialized 2011 English translation of the *Iliad*[143] and Alexander Pope's controversial English translation of Homer in the early 18th century, to the French translation of the *Iliad* for François I composed by Salel, and even the early partial Latin

[141] Plato was one of Homer's earliest critics.
[142] Some of the more popular Linux desktop environments include Ubuntu, Debian, Red Hat, and Fedora.
[143] Homer. *The Iliad.* trans. Stephen Mitchell. New York: Atria Books, 2012.

translations of Homer used by Petrarch, can all be viewed as distributions of Homer.

The parallels – or perhaps, in the spirit of romanticism, they should be called metaphors – that I describe when comparing Linux and Homer might be nothing more than my own willingness to succumb to the temptation of technological gimmickry. I certainly would not be the first person tempted to use contemporary technologies as metaphors for understanding complex systems. For example, the currently popular metaphor for understanding the human nervous system is the model of computers, but less than 200 years ago the popular understanding was the model of Industrial Revolution mechanical machines.[144] Regardless, my intention is not to redefine well-established humanist artistic production technologies as software. I propose the reverse. I intend to understand software as a new form of humanist artistic production technology. Doing so enables the unique opportunity of experiencing, firsthand, the ways in which an artistic medium's earliest traditions and processes are established. With that experience comes an opportunity to revisit previously-established artistic traditions that have become so standardized we can easily forget how they, too, were once new technologies.

The case of comparing the development of Homer and the development of Linux offers a tantalizing possibility. The community-developed nature of Linux means the version of Linux you might have

[144] As Jaron Lanier reminds us, "Before computers came along, the steam engine was a preferred metaphor for understanding human nature" (Location 654).

running on your cell phone or that your favorite website might have running on its web server is not the same Linux as was originally created by Linus Torvalds. Nor is that version of Linux likely to tell you much about Torvalds himself. But that Linux, which you might be using right now or will likely be using in the very near future, contains traces – some subtle and some not-so-subtle – of the original Linux. Presumably, if it were possible to gather enough versions of Linux, both contemporary versions and older versions, and if it were possible to access the distributed version control histories that evolved in conjunction with Linux, it would be possible to uncover not necessarily the original Linux in its entirety, but enough of the original Linux to provide a strong understanding of both the historical context in which that original Linux operated and perhaps even some understanding of the man who created it.

If that possibility exists for Linux, then perhaps the same is true for Homer. If it were possible to gather enough versions of the Homeric texts (i.e. its many translations), and if it were possible to access the distributed version control histories that evolved in conjunction with Linux (i.e. all of the Homeric scholarship), then it may be possible to uncover not necessarily the original Homer in its entirety, but enough of the original Homer to provide a strong understanding of both the historical context in which that original Homer operated and perhaps even some understandings about the person (or people) who created it.

The Homeric question seems to be precisely this kind of comprehensive information-gathering pursuit. It is a pursuit largely recognized as untenable, but its impossibility is only a secondary concern because the uncovering of the original Homer – the Homeric Question – is not about Homer at all. The Homeric Question is, as thousands of years of Homeric development clearly indicate, the Homeric software's next cultural iteration. As such, the Homeric Question is not really "Who was Homer?" The Homeric Question we should consider asking ourselves is: "Why does current Homeric scholarship care so much about recovering a sense of the original Homer?"

Chapter 3: There Is No World Without Verona Firewalls

At 12:34 AM on Thursday, November 8, 1988, Andy Sudduth, systems manager at Harvard University's Aiken Laboratory, posted a message to an Internet bulletin board. The message's brevity belied the importance of its content:

There may be a virus loose on the internet.

Sudduth's post may or may not have been entirely correct; the question of whether or not what Sudduth was witnessing was a computer virus or a computer worm remains a contentious subject.[1] Regardless of the technical classification, Sudduth was alerting Internet users to an unprecedented malicious digital attack. The attack would eventually be known as the Morris Internet Worm, and it is remembered primarily because the Morris Worm was one of the first malicious coding exploits to propagate across computers through something other than infected hard media storage disks. As a result, the Morris Worm became, proportionally, one of the most devastating malicious computer programs (malware) to ever spread online.

[1] In a technical memo describing the event soon after, Purdue University Professor Eugene H. Spafford explains the confusion, writing: "There seems to be considerable variation in the names applied to the program described in this paper. I use the term *worm* instead of *virus* based on its behavior. Members of the press have used the term *virus*, possibly because their experience to date has been only with that form of security problem. This usage has been reinforced by quotes from computer managers and programmers also unfamiliar with the terminology…. The program that was loosed on the Internet was clearly a worm" (3).

Sudduth's alert was caused by a 23-year-old, first-year doctoral student at Cornell University named Robert Tappan Morris. Morris had created a computer program designed, according to him, to determine the size of the Internet. Media reports at the time explain that Morris launched the virus and then went to dinner. When he returned, he discovered that an error in his code was causing his program to behave in an unexpectedly damaging way. Instead of just embedding itself automatically on host machines across the Internet and reporting minimal statistical data, as Morris allegedly intended, an aspect of the code designed to overcome errant redundancy checks was causing the program to install itself on the same machines multiple times. Morris's error caused the program to overwhelm the processing resources of infected computer systems with requests to operate its multiple installs, making those computers unable to perform other tasks. The result was devastating. Estimates during the immediate aftermath of the outbreak indicate that Morris's program caused the shutdown of 6,000 computers, which, at the time, accounted for approximately 10% of the Internet.[2]

The only person who knows the actual reason for Morris's program is Morris himself,[3] but a wide variety of reasons have been suggested, including poor programming skills, an accident related to drowsiness, or an attempt by Morris to alert the Internet community of largely ignored security risks. Whatever the actual reason, Morris's program functioned as a Distributed

[2] This estimate of the Morris Worm's scale is somewhat controversial, but all evaluations of the event agree that, whatever its exact effect, a large and unprecedented amount of the Internet's computer infrastructure was maliciously taken offline in a brief amount of time.
[3] Today, Morris is a highly regarded technology investor and tenured professor in the department of Electrical Engineering and Computer Science at MIT.

Denial of Service (DDoS) attack. A DDoS attack is a type of network vulnerability hack intended to disable a computer by overwhelming its resources through excessive data requests. Due the extreme number of requests and the processing resources required to serve them all simultaneously, the under-siege computer system ultimately fails. In the case of the DDoS attack caused by Morris's program, the young grad student revealed, in a highly public way,[4] a critical weakness of fledgling Internet technologies. The world's exciting, new, globally connected computer network was not prepared to protect itself against the very security vulnerabilities its connectivity enabled.

In retrospect, the Morris Worm was, for an underprepared Internet community, a driver of beneficial change. Consider the Internet's then-primary users. At the time, the Internet was primarily used for hosting and distributing government and university research information. While the Morris Worm could have stolen or destroyed important data, all it did was cause the temporary shutdown of a few thousand computers. By shutting down those computers, Morris demonstrated the potential of someone far less ethical than him stealing sensitive government information and research data. This revelation gave the pioneers of the Internet an important warning about security. Their powerful new technology for information storage and dissemination required them to be concerned with more than just how to use

[4] The Morris Worm was the first computer virus/worm to attract major public and media attention. The substantial press coverage it received demonstrates its prominence particularly because the press coverage came during a US presidential election year, and only a few days before the election itself.

the Internet to preserve and access information; they also needed to develop security mechanisms in order to prevent malicious data corruption.

While the specific lessons of the Morris Worm are mainly limited to a fledgling storage and transmission technology – the Internet – directly affected by its malevolent actions, considering the Morris Worm in relation to both current and past information technologies suggests additional, more universal implications. Since, as previously noted, the Internet was not the first global technology for information storage and dissemination, surely the Morris Worm was not the first malicious disruption of such a technology. Recognizing this likelihood allows the lessons of the Morris Worm to be used as instructive analogies about how the discovery of and response to technology vulnerabilities contributes to an information technology's development trajectory. For example, the response to the Morris Worm was not to permanently shut down the Internet. Instead, the Internet's earliest pioneers strengthened their new technology of information storage and dissemination by engineering solutions to prevent similar security breaches.

Just as the Internet faced (and continues to face) security challenges, other information storage and dissemination technologies, including technologies like literature, likely had (and still have) stability and security vulnerabilities. Because of those security vulnerabilities, they were likely – to appropriate a digital term – "hacked." And because they were hacked, and because, despite being hacked, they continue to persist and flourish, the history of those technologies also likely include fixes and updates in order to better maintain stability and protect information.

***

If the existence of protective mechanisms for other information storage and dissemination technologies like literature, art, and music seems unlikely, consider that the protective technologies of the Internet are largely unnoticed by its day-to-day users. Most users are familiar with basic security technologies, like password protection. But, when entering credit card information, how many users recognize (or care) if a website is using an SSL certificate?[5] When providing a website with a social security number, how many users consider whether or not that website is preventing cross-site scripting attacks?[6] When sending an email containing a bank account password, how many people first consider whether their email system is being managed by an email provider using transport layer security?[7]

The security mechanisms embedded in information technologies are, by their nature, designed to avoid the user having to think about them. As a result, using a secure website, submitting private information, sending a secure email, or leveraging any other information technologies to store and transmit information rarely forces the technology's user to consider how such technologies are being secured. If subtlety and unobtrusiveness are core

[5] An SSL Certificate is a cryptographic document that, when used in conjunction with websites, helps to verify that the website is the site it is claiming to be for the user, and it prevents other people from eavesdropping on the data being exchanged by users' computers and the site's servers.
[6] Cross-site scripting (XSS) is an exploitable software vulnerability, primarily related to websites and the World Wide Web, that allows malicious client-side code to be injected into people's computers without them or the host site being aware of it.
[7] Transport Layer Security is a more secure iteration of the same Secure Sockets Layer (SSL) technology used in SSL certificates to prevent unauthorized eavesdropping and access.

elements of digital security mechanisms in order to avoid interfering with the primary purpose of the technology itself, is the same true for non-digital information technologies? Reading a book, listening to music, watching a play – even if the users of these kinds of humanist technologies are not aware of how their information is being secured, are mechanisms in place to ensure data is being safely stored and safely transmitted?

The general cultural familiarity of language described in my first chapter suggests that one of the biggest obstacles to answering these kinds of questions about the information security protocols of literary technologies is likely the challenge of overcoming platform standardization. Platform standardization occurs when the variants of a technology coalesce around a single operating and/or design paradigm. Such standardization has its benefits in terms of technology adoption and transmission. For example, a computer's QWERTY keyboard has become a standardized technology, making it easy for the user of one computer to type on another computer. However, despite its benefits, standardization can delay or prevent improvements because users become so familiar with one way of using a piece of technology that they are either unwilling to change to an alternate implementation, or, in many cases, might not realize alternate implementations exist.[8]

[8] In the case of the QWERTY keyboard, for example, it was designed specifically to slow typing speeds on early typewriters because their mechanical parts could not accommodate typing speeds on alternate key configurations. Variants of the keyboard are available today, such as the Dvorak keyboard, which reportedly enable faster typing speeds. However, people have become so accustomed to the QWERTY layout that, despite its inefficiencies, it is still, by far, the predominant keyboard configuration, including simulated keyboards on touch screens where any concern about moving parts is irrelevant.

This kind of standardization is rampant in long-established technologies, with the technology of books being one prominent (and complex) example.[9] As a result, the security mechanisms are, like the technologies themselves, so deeply embedded in standard societal and cultural usage that recognizing them as anything other than the technology's logical and original structure becomes an analytical challenge. In contrast, digital information technologies are yet to be as standardized and culturally normalized. By living during an age when an influential global technology of information storage and dissemination has yet to be fully coupled with many of the security mechanisms that will ultimately help it combat data corruption, digital scholars have an opportunity to witness how such security mechanisms evolve. Perhaps, as we witness the development and implementation of digital security, we can use the information gained from watching that process to better understand the development of previously established information technologies by employing the technology of metaphor. Metaphor – itself a sort of analytical technology[10] – provides a way to envision hidden information security mechanisms in literary texts by mapping the processes through which digital security mechanisms evolve

[9] The transition from manuscript to book culture, as well as its hurdles and results, is famously considered in Elizabeth Eisenstein's *The Printing Press as an Agent of Change*. The current transition from paper books to e-readers seems poised to affect similar change. Being in the midst of the transition, we cannot know its ultimate results, but the early hurdles to e-reader adoption have already been seen. Readers are so comfortable with the technology of the book that many have resisted the idea of separating the content of the book from the physical, printed object.

[10] The use of metaphor as a technology for understanding complex concepts is a theory popularized by George Lakoff and Mark Johnson in *Metaphors We Live By*. They explain that, "metaphor is pervasive in everyday life, not just in language but in thought and action. Our ordinary conceptual system, in terms of which we both think and act, is fundamentally metaphorical in nature" (3).

onto what appear to be similar phenomena occurring during significant development stages of literary technology.

Before initiating that metaphoric mapping process, I should note that the topic of digital security is, itself, so complex that no single chapter, or single book, would be able to address it with any level of appropriate comprehensiveness. But the question of whether or not such security mechanisms exist is easy to answer: of course they do. For the purposes of this project, that answer is sufficient.

Recognizing the existence of digital security mechanisms is easy because the newness of the technology has made the development of its security protocols a similarly new phenomenon. However, 1,000 years from now, when digital technologies are so ingrained in cultural operations that they are hardly recognized as anything other than normal, or perhaps even considered "antiquated," how obvious will digital security mechanisms be? The same problem – a problem of standardization – could be making the existence and nature of textual security protocols difficult to recognize. Because of this difficulty, I propose approaching the topic of data security in literary technology by analyzing important points of failure.

By considering a perspective of failure, I mean to suggest that the best way to unearth literary security protocols might be by avoiding direct questions like: "What are the security protocols within literary technology that prevent malicious data corruption?" Instead, the "failure" approach I am suggesting is visible when considering the example of the Morris Worm. When Robert Morris created what would become known as the Morris Worm,

he never intended to develop a comprehensive understanding of the Internet's security logistics. Instead, Robert Morris exploited the Internet's security vulnerabilities. However, by doing so, Morris unintentionally spurred development of many of today's most critical pieces of Internet security infrastructure.

Because the Morris Worm's exploits of then-current technology propelled security-related innovation, the ability to understand how the Morris worm influenced digital security offers a unique analytical parallel to how security may have developed in literary technology. For example, if confronted with a situation of having to extrapolate digital security mechanisms of the Internet without knowing exactly what those security measures are, studying the exploits of the Morris Worm would produce general information about digital security mechanisms. By studying the ways the Morris Worm breached the technology of the day, and by knowing that those exploits were subsequently either corrected entirely or, if complete correction was not possible, significantly improved upon, we could extrapolate what kinds of security protocols were created.

What if we apply the same technique to understanding security mechanisms within literary technology? If an exploitative event similar in impact to that of the Morris Worm's impact on Internet security occurred during earlier stages of literary development, could processes and tools for data protection have been developed and implemented in order to prevent such an attack from happening again? And if such processes had been

created, could analysis of the event's exploits help cultivate an understanding of embedded literary security mechanisms?

Finding an exploitative event in literary history with a similar kind of impact as that of the Morris Worm begins by first understanding what differentiates Morris's impact from that of other digital influencers. In the same way that numerous novelists, poets, and dramatists have impacted the evolution of literary texts and textual security mechanisms, numerous genres of digital developers have significantly influenced digital technologies in a variety of different ways. For example, by 1988, when the Morris Worm was causing a massive shutdown of websites, Bill Gates' soon-to-be ubiquitous Windows operating system had been publicly available for purchase and use for three years. A year before Gates launched Windows, Steve Jobs had launched his first industry-changing Macintosh computer system. Clearly, before the Morris Worm forever changed digital technology, other impactful software developers substantial influenced the digital age. What makes the Morris Worm both unique and beneficial for discovering the evolution of data protection protocols and procedures is not simply the extensiveness of Morris's impact, but also the way in which Morris engaged his chosen medium: Morris attacked the technology's weaknesses. In contrast, digital visionaries like Gates and Jobs were self-consciously attempting to improve and expand digital technologies. Morris was not developing software to improve digital technologies. Instead, Morris achieved his fame (or infamy, depending on one's perspective) by exploiting existent technological vulnerabilities. Robert Morris was a hacker.

The "hacker" label is, within the technology community, highly contentious. Mass media appropriation of the term gives the label a negative connotation by associating it with malicious intentions. Hackers, according to the vernacular of the mainstream media, are a technologically proficient community that uses its knowledge of technology to exploit vulnerabilities in digital systems, often with goals of disruption, theft, or malicious self-indulgence. However, the origins of the term (at least in a digital sense) are embedded in academia and early digital pursuits at MIT. The hacker culture is not maliciously destructive; the hacker culture is concerned with deconstructing systems in order to better understand how they operate. Sometimes the goals of this deconstruction are productive, and a hacker is either attempting to rebuild the system more efficiently or build something entirely new from its pieces and the pieces of other similarly deconstructed systems. Other times, the goal is simply to break things apart and see what happens.[11]

Not unexpectedly, general hacker culture takes offense to the popular, media-driven interpretation of the hacker label that describes all hackers as people with malicious intent. Non-malicious hackers prefer to call the people who exploit security protocols "crackers."[12] By giving someone like Robert

[11] In a discussion of coding protocol as it relates to web development, Thomas Rickert notes that "hackers are ultimately drawn to the possibility of exploiting protocol and making it work otherwise" (L 439). Rickert is drawing on the work of media theorist Alexander R. Galloway who argues, "What hacking reveals [...] is not that systems are secure or insecure, or that data wants to be free or proprietary, but that with protocol comes the exciting new ability to leverage possibility and action through code" (172).

[12] Even the exploitative hackers are not necessarily so easily defined as malicious. Within the sub-genre of security hackers (or "crackers") are divisions such as "White Hat" and "Black Hat." Black Hat hackers are the ones associated with malicious intentions, while White Hat hackers are

Morris the label of "hacker", I deploy it here as a neutral reference. I apply it to Morris in order to convey the uncertainty around his intentions. Morris claims his intentions for creating and releasing his ultimately destructive program were experimental and intellectually motivated. However, at the time, Morris was the focus of the first major media coverage of a digital security breach, he had been responsible for, by some estimates, $10,000,000 worth of damages, and he was being prosecuted under the Computer Fraud and Abuse Act.[13] If I were faced with such circumstances, I suspect I, too, would loudly proclaim my intellectual curiosity as the reason for my exploits to anyone who bothered listening.[14]

For my purposes of leveraging Robert Morris and the Morris Worm as a model for how to recognize security protocols within a connective technology that specifically prioritizes hiding them, Morris's intentions are not as important as his actions and their results. The process of building and releasing the program that would eventually bear his name was a hacker pursuit. Morris exploited a structural flaw within a well-established system, ultimately revealing a weakness in that system and enabling a reconstruction of the system in a more sustainable way. Finding a literary parallel to Morris means finding an author who, instead of creating new texts, intentionally exploits the flaws in well-established literary systems. By doing so, that author

---

usually (but not always) professional security experts who search for vulnerabilities in order to report them and have them fixed. To further complicate the term, "Grey Hat" hackers operate on an ethical level between White and Black Hat hackers, with their motives not necessarily being illegal, but their processes often including elements of questionable legal acceptability.

[13] Morris was the first person to be prosecuted under the Act. His case would set numerous legal precedents for how the federal government would deal with malicious computer intrusions.

[14] Suspiciously, although Morris was a student at Cornell, he programmed the worm so that it appeared to be released from MIT, suggesting some sort of mal-intent.

– that hacker – is ultimately helping reconstruct those systems in more sustainable ways. According to that criteria, I wonder if the author whose textual hacks are most useful for uncovering the security mechanisms hidden within literary technology is the author who, in terms of influence, could be described as the most prominent literary hacker in history: William Shakespeare.

If my labeling of Shakespeare as a hacker seems like a tenuous claim, I suggest its tenuousness is the product of Shakespeare's immense hacking skills. The famous bard from Stratford may have been so successful at hacking literature and then reconstructing it in a more efficient and more sustainable way that readers continue to live in Shakespeare's hacked literary tradition. Escaping this literary tradition – if such an escape is desirable or even possible – would not be easy. But for the purposes of uncovering the security mechanisms helping to prevent data corruption within literary technology, escaping Shakespeare's literary tradition may not be necessary. Instead, I propose using a Shakespeare text in the same way someone might use the Morris Worm to distill information about Internet security mechanisms. In the case of the Morris Worm, knowing what technological weaknesses the worm exploited, and then knowing that those weaknesses were corrected in response, reveals the kinds of security mechanisms subsequently added to the Internet in order to prevent a similar future attack. In the case of Shakespeare, knowing what Shakespeare exploited, and then knowing those weaknesses were corrected in response, might present a similar opportunity

to infer the security mechanisms developed in response to Shakespeare's hacks. Those literary security mechanisms, like the digital community's response to the Morris worm, would have been created to prevent the same weaknesses from being exploited again.

My theory that a Shakespeare text can be used to extrapolate information about literary security mechanisms in the same way the Morris Worm can be used for understanding digital security relies on scale. In the case of the Morris Worm, while other malicious programs have continued exploiting digital vulnerabilities, few other attacks of a similar magnitude in terms of both actual destruction and perceptions of destruction relative to the then-current scale of the technology have happened since, which is why the Morris Worm is often referred to as "The Great Worm." Because no security exploits have the relative scale of the Morris Worm's digital devastation, any analysis of digital security that leverages knowledge of the Morris Worm benefits from knowing that the most critical technological vulnerabilities the worm exploited are, to this point in the Internet's short history, secured.[15] Knowing that those weaknesses are secured supports subsequent theorizing about what those security mechanisms actually are. In other words, because we know that the level of devastation caused by the Morris Worm has not occurred again, we also know we could analyze how the Morris Worm exploited vulnerabilities in the Internet and, without knowing the exact

[15] Internet security vulnerabilities are not necessarily limited to being application-level vulnerabilities. While more potent and more malicious software has affected the Internet following the Morris Worm attack, the Morris Worm instilled in the technology community a powerful sense of awareness about the possibility of security threats. Though not directly software or hardware security mechanisms, awareness and vigilance are two powerful protective barriers that have provided a significant security layer to most digital transactions.

technologies implemented to prevent such an attack from recurring, we could hypothesize what the nature of those technologies are.

Consider the following example of using a specific Morris Worm exploit to deduce the kinds of security mechanisms it ultimately fostered. One of the ways in which the Morris Worm spread to other computers was through a vulnerability in operating systems that allowed the worm to get basic information about a computer's users, such as names and usernames, from a system's `/etc/passwd` file. From that file, the Morris Worm compiled possible passwords derived from user information, such as a user's initials or a user's combined first and last names, and then the worm attempted to use those possible passwords in order to gain access to the system's secured files. If a proper password was found, the Morris Worm would search the system's `.rhosts` and `/etc/hosts.equiv` files and use information from those files to determine other potential systems around the network on which to embed itself, which is what helped the Worm spread across the Internet. Without knowing the specific security mechanisms implemented to protect against that particular exploit, but by knowing the nature of the exploit, and by knowing that no "cracker" has deployed an exploit on any similarly large (relative) scale, we can readily infer the kinds of security mechanisms used to prevent the problem in the future. In this particular case, we can infer that some sort of mechanism was implemented to prevent external programs from accessing a system's `/etc/passwd` file containing user information, we can infer that password input attempts were limited in order to limit the possible

success of "brute-force" attacks,[16] and we can infer that, on a social level, Internet users have become more aware of the need to create more secure and more complex passwords.

I do not mean for the above example to be a comprehensive explanation of how analyzing a software hack like the Morris Worm becomes a foundation for logical inferences and hypothesis about current digital security processes. Instead, the example should be illustrative. I could develop more complex and technical theories about security mechanisms derived from the exploits of the Morris Worm, but developing those theories would be an unnecessary exercise since the event is recent, as are records and memories of it and the responses to it. In contrast, in the case of a suspected literary hack that might have occurred hundreds or even thousands of years ago, similar records, if they ever existed, are unavailable, meaning the only resources readily available for unearthing security mechanisms deeply embedded in the technology of literature are inferences and hypothesis. To better generate those kinds of resources – to better generate logical guesses about the security mechanisms embedded in literary texts that help prevent data corruption – I propose looking toward literary equivalents of the Morris Worm by considering texts with lasting impacts, intended or unintentional, on subsequent literature.

Constructing an argument about an exploitative literary hack and its influence on the security technology of literature would be easier to create if

[16] "Brute-force" attacks are a type of cryptographic exploit involving checking all possible security keys until the correct one is found.

that literary hack had the same kind of system-wide influence as the Morris Worm. Because the Morris Worm did so much damage to the existing technology system, and because the Morris Worm garnered so much attention and publicity, it was able to influence the Internet's security mechanisms in ways that a less potent and a less public attack likely would not have. For example, the event received so much media coverage at the time that even people with only a minimal understanding of the Internet became aware of the necessity of protecting it against data theft and corruption. One author whose texts unequivocally have the same kind of impact on literature, both within the industry of people producing literary technology as well as with people only vaguely familiar with using literary technology, is William Shakespeare. Because so many people around the world recognize the name "William Shakespeare", because so many people have a sense of who Shakespeare was regardless of their relationship to Shakespeare's work, and because of the scale of Shakespeare's impact on literature and literary production, if literary technology has security mechanisms for protecting its data, then Shakespeare's texts were likely playing a role in developing some of those mechanisms. Uncovering those aspects of literary security, as a result, begins with William Shakespeare, one of literature's most recognizable names.

***

"What's in a name?" young Juliet famously asks in Shakespeare's *Romeo and Juliet* (II.ii.43).[17] Her question applies most immediately to the name of Romeo Montague, a name which is the only thing seeming to separate Juliet from her newly discovered love. However, Juliet never has time to fully consider the social complexities intimately bound within naming conventions that could help her (and her audience) properly understand both the importance of her question and the consequences of acting according to an incomplete answer. Instead, Romeo interrupts her contemplation of the question through his eagerness to overcome the physical barrier of a name – an easy task – without considering the more constricting social barriers. If Juliet had the opportunity to assess what exactly is "in a name," perhaps her story, and Romeo's, would have ended less tragically.

Shakespeare's play seems to realize that the complexities inherent in names and naming conventions are not only substantially more difficult to ignore than the young lovers want to believe, they become, appropriately or not, the containing walls in which ideas are encapsulated, processed, and distributed. Note that, in order for Romeo and Juliet to overcome their names, they must first overcome the physical wall that surrounds the Capulet house, a wall Romeo circumvents "with love's light wings" (II.ii.66). However, by flying over the Capulet wall, Romeo does not destroy the wall, meaning it still remains, both as a physical divide between Capulets and Montagues, and as a metaphorical divide between their conflicting ideologies and belief

[17] In the spirit of this project, I elected to use a digital version of *Romeo and Juliet* in order to explore whether such versions might contain the kind of quality necessary for advanced critical inquiry. All textual citations of *Romeo and Juliet* will be derived from the following edition: Shakespeare, William. *Romeo and Juliet.* ed. Joseph Pearce. San Francisco: Ignatius Press, 2011.

systems. The result is that Romeo and Juliet never circumvent the wall created by their conflicting family names. Instead, the securing wall does its job by ultimately, and tragically, preventing Romeo's attack from succeeding.

The central lesson of Romeo and Juliet's story, at least on this reductive level, is a lesson about the power affixed to names. Shakespeare's play demonstrates how ignoring the sociological and cultural associations embedded in established and well-known names is ultimately a poor decision that, while not often ending in actual death, as in the case of *Romeo and Juliet*, usually ends in some form of defeat or failure. Here, again, the play begins this argument for its audience when Juliet notes that "a rose / By any other name would smell as sweet" (II.ii.43-44). Noticeably absent from Juliet's famous thought is that regardless of whether or not a rose with any other name would smell as sweet, because social convention has conferred upon the object the name of "rose," calling it something else when referring to the flower in its absence – for example, trying to convey the concept of a rose but using the word "carrot" – would disseminate errant information.[18]

In the absence of the physical objects they describe, names must represent their objects and convey all necessary information about them in order to generate common meaning. A name must be both metaphor and metonymy at the same time, two rhetorical opposites combined into one linguistic construct, allowing such devices to serve as both representative

[18] In an interesting parallel, when presenting the argument concerning whether the malicious code should be labeled a "virus" or a "worm," one analysis of the Morris Worm ("With Microscope and Tweezers: Analysis of the Internet Virus of November 1988" by Mark W. Eichin and Jon A. Rochlis) titles the section "A Rose by Any Other Name." This nod toward Shakespeare, intentional or not, underscores the importance of naming conventions and the kinds of information transmitted through names.

substitution and substitutive abstraction. By appropriating this understanding of the inherent complexity of a name, the answer to Juliet's question about "what's in a name?" becomes equally complex: names have the power to convey highly complex shared information schemas in extremely concise and efficient information storage mechanisms, and, by doing so, they empower a technology for disseminating complicated sociological event data in an efficient and effective manner so that large population groups can more easily understand, recognize, and respond to information. Described more succinctly: a name contains everything a person needs to know about a topic to foster mutually-established baselines of understanding for clear and effective communication.

By properly considering the complexities of names, their inherent power to simply and effectively control meaning appears astounding. The capacity of the name "rose" to convey to millions of people enough information to establish a shared, detailed understanding of the object being discussed without any additional description – plant, flower, bud, stem, petals, thorns, etc. – gives names an influence over societal operations and interactions that, when properly manipulated, suggests that the ability to dramatically effect naming conventions could be a valuable skill for a writer.

The significance of names in the Romeo and Juliet story, and, specifically, names at odds with one-another, has origins long before the story took the names "Romeo" and "Juliet." The legend centers on two embittered families, each represented, both literarily and for the characters within the story, by two names: Montague and Capulet. The juxtaposition of the

Montague and Capulet names appears to have its roots in the sixth canto of Dante's *Purgatorio*. Dante, in an attempt to alert the uncaring emperor Albert of Hapsburg to the kind of civil strife engulfing Italy, writes:

> Vieni a veder Montecchi e Cappelletti,
>
> Monaldi e Filippeschi, uom senza cura;
>
> Color già tristi, e questi con sospetti.[19]

In a strange twist of history, the remaining evidence of the Montecchi and Cappelletti suggests that not only were the two factions not in direct opposition with each other, they were not even in the same city. The Montecchi seem to, as the Romeo and Juliet story indicates, come from Verona, but the Cappelletti appear to be from Cremona. Olin H. Moore, in his book chronicling the origins of the famous love story, *The Legend of Romeo and Juliet*, explains the most convincing theory for why the two factions were literarily juxtaposed as rival families in the story that would eventually become *Romeo and Juliet*. According to Moore, the obscure passage in *Purgatorio* is to blame for the simple reason that "Dante balanced the name of the *Montecchi* and *Cappelletti*, factions which had destroyed themselves in past strife (*color già tristi*) with the contemporary and better known parties – the Monaldi and Filippeschi of Orvieto – who were in a dubious situation because of their dissensions (*e questi con sospetti).*"[20] If Moore is right, then the famously hostile names of "Montague" and "Capulet" are only famously hostile because Dante needed to balance his poetic line, and, by doing so,

---

[19] Dante. *La Divina Commedia*. ed. D. Lodovico Di Borbone. 1796. 35.
[20] Moore, Olin H. *The Legend of Romeo and Juliet*. Columbus, OH: The Ohio State University Press. 14.

placed two non-dissenting historical Italian factions in parallel with two then-contemporary dissenting Italian factions. Readers at the time errantly assumed the juxtaposition meant the Montecchi and Cappelletti were also at odds with each other, and from that errant interpretation, the famous households of *Romeo and Juliet*'s took their names.

While the names of the *Romeo and Juliet* households likely have their origins in Dante, the *Romeo and Juliet* story itself appears to have its roots in the very real tendency of historical Italian city-states to have armed conflicts between quarreling families. These battles eventually began to be reenacted in stories of the time, with the most prominent being the 13th century Italian story of *Ippolito e Leonora*. Ippolito and Leonora are the son and daughter of the warring Buondelmonte and Bardi families, respectively, they meet at a party, fall immediately in love, are devastated to discover one-another's familial ties, and secretly marry before Ippolito is arrested as a thief while trying to visit his bride. Although sentenced to death, Leonora prevents the execution by publicly announcing her love for and secret marriage to Ippolito, a gestures which ultimately reconciles the families.[21]

The first imposing of the *Montecchi* and *Cappelletti* names on the "warring Italian households" story appears in 1530 when Luigi da Porto publishes his first edition of "Giulietta e Romeo."[22] Most scholars tend to cite this as the original formulation of what would ultimately become

[21] For a more detailed summary of the *Ippolita e Leonora* story and its relationship to *Romeo and Juliet*, see chapter IV of Moore's *The Legend of Romeo and Juliet* (29-33).
[22] Although, in 1530, da Porto appears to be the first person to append the *Montecchi* and *Cappelletti* names to the *Ippolita e Leonora*-style love story, Francesco da Buti, in 1380, appears to be the first person to reference the supposed hostility between the two families (Moore, 29).

Shakespeare's story, and it certainly appears to be the origins for the coming together of the story's most famous names: Romeo, Juliet, Montague, and Capulet. But what might be most interesting about this evolutionary process toward the famous names is how the story progresses prior to the standardizing of names. I have already noted some of those steps as they relate to Dante, famous Italian households, and the *Ippolito e Leonora* tale, but the *Romeo and Juliet* story has a developmental history that draws from a wide array of historical texts and ideas ranging from ancient Greece (a likely product of the Homeric technology discussed in my second chapter), to Ovid's *Metamorphoses*, and Boccaccio's *Decameron*. Recognizing these origins, even if their exact details are ambiguous or unknowable, helps in, to return to Shakespeare's story, answering Juliet's question of "what's in a name?" Before the *Romeo and Juliet* title could embody the complexity of what has become Shakespeare's familiar and famous love story, its core components had to undergo numerous iterations and reformulations. The Romeo and Juliet story had to reach a certain level of cultural maturity, standardization, and acceptance before it stabilized and received a single name that explained the entirety of the story's core concepts. As a result, just as the name "rose" generates in the minds of millions of people a stable concept of a particular kind of flower on which further conversation about the flower can be built, so, too, can a story named "Romeo and Juliet." Despite the "Romeo and Juliet" title being linkable with da Porto, as well as Bandello, Boaistuau, Arthur Brooke, William Painter, and perhaps others who may have appropriated the

story and its names in one form or another throughout its development,[23] the "Romeo and Juliet" name appears to belong to the person who stabilized it and made it a common element on which to build future discussions and iterations of the story – William Shakespeare.

The association of the "Romeo and Juliet" name with Shakespeare, as opposed to the numerous other people who called a very similar and, at the time, very popular story by a nearly identical title, demonstrates what Foucault calls the "author function." Foucault explains that:

> An author's name is not simply an element of speech (as a subject, a compliment, or an element that could be replaced by a pronoun or other parts of speech). Its presence is functional in that it serves as a means of classification. A name can group together a number of texts and thus differentiate them from others. A name also establishes different forms of relationships among texts.[24]

The name Shakespeare, as Foucault notes, represents more than a singular person. In the case of *William Shakespeare's Romeo and Juliet*, it seems to represent a literary history that both precedes and proceeds the personhood (or possible personhood) of the author by hundreds of years. And yet, the story was is mostly associated with Shakespeare. Why is the famous tale of

---

[23] Arthur Brooke famously makes reference to a now-lost, dramatic version of the story that would have preceded Shakespeare's. Brooke writes: "Though I saw the same argument lately set foorth on stage with more commendation, then I can looke for: (being there much better set forth then I haue or can dooe) yet the same matter penned as it is, may serue to lyke good effect, if the readers do brynge with them lyke good myndes, to consider it. which hath the more incouraged me to publishe it, suche as it is" (Moore, 95).

[24] Foucault, 123.

young lovers from warring Verona families commonly referred to as Shakespeare's *Romeo and Juliet* instead of Brooke's *Romeus and Juliet* or da Porto's *Giulietta e Romeo* or any other more senior version of the tale? What does Shakespeare do to the story to make his version the defining version of the social construct we now call "Romeo and Juliet" as opposed to any of the other versions? Conversely, we might also benefit from asking: What do we do to Shakespeare?

In one prominent example of this modern proclivity for defining *Romeo and Juliet* as a Shakespearean construct, as opposed to a culturally mutable narrative framework, Baz Luhrmann's 1996 screen adaptation of the story is titled *William Shakespeare's Romeo + Juliet*. However, much of Luhrmann's movie pulls from sources extending far beyond the realm of Shakespeare. Those sources range from the American Western genre to the moment when Juliet wakes mere seconds before Romeo's death, a dramatic climax that occurs in Brooke's version of the story but not Shakespeare's.[25] While much of the plotline and characters use Shakespeare's language, Luhrmann's visual style is certainly unique, and, for mediums so intensely entangled with the visuality of events – both Shakespeare's stage-based performance and Lurhmann's film version – surely the movie is as much "Luhrmann's Romeo + Juliet" as it is Shakespeare's.

[25] For more discussion of how Luhrmann's film is compiled of sources beyond Shakespeare, see Courtney Lehmann's "Strictly Shakespeare? Dead Letters, Ghostly Fathers, and the Cultural Pathology of Authorship in Baz Luhrmann's *William Shakespeare's Romeo + Juliet*," Philippa Hawker's "DiCaprio, DiCaprio, Wherefore art though, DiCaprio?", and Barbara Hodgdon's "*William Shakespeare's Romeo + Juliet*: Everything's Nice in America."

I realize questioning what makes Shakespeare's version of the Romeo and Juliet story the culturally venerated version might, at first, seem unfairly simplistic and ultimately untenable. Such questions can appear concerned with little more than attempting to distill the talents of Shakespeare through a basic comparison of his work with (possibly) inferior texts. I will not be pursuing such a study. I am not questioning the dominance of Shakespeare's appropriation of the Romeo and Juliet story for the sake of embarking on a line-by-line comparison of Shakespeare's *Romeo and Juliet* with all the versions that either preceded it or came after. Instead, I ask the questions because the existence of so many versions of the same basic story is an example of literature functioning as a cultural information technology being regularly iterated and improved upon in order to better convey standard sets of cultural data. As a result, asking such questions does not need to be an exercise in making retroactive, opinion-based judgments about why one literary decision is better than another. Instead, asking questions about why Shakespeare's version of the Romeo and Juliet story has become the culturally standard version allows for an opportunity to explore how Shakespeare may have hacked the weaknesses in a famous literary tradition and ultimately rebuilt that tradition in a way that improves its ability to preserve and disseminate information.

Appropriating the term "hack" to describe Shakespeare's approach to the Romeo and Juliet tradition implies that the tradition itself is an information technology capable of being hacked, and that hacking it would

reveal some sort of vulnerability in the ways in which that technology secures its information. If Shakespeare does indeed hack the Romeo and Juliet story, then the first task (for both Shakespeare and ourselves) is to recognize the underlying technology that makes such a story possible. With the title of his book, *The Legend of Romeo and Juliet*, Olin Moore gives the technology the name of "legend." Alternate names include "myth" and "tragedy." But, from a technological perspective, an article by Lawrence Edward Bowling might have the most useful description. The title of Bowling's old but well-articulated article is "The Thematic Framework of *Romeo and Juliet*," and in it, Bowling makes the argument that:

> In its broadest terms, *Romeo and Juliet* deals with the wholeness and complexity of things, in contrast with a partial and simple view. This theme is functional on various levels in almost every speech and action in the play; almost every character is at some time shown discovering that some particular thing has not merely one quality which is pure and single but many qualities which are diverse and multifarious. This theme often comes out in the form of paradox, since the characters are constantly being shocked and surprised to discover that one quality of a thing is often in sharp disagreement with some other quality.[26]

[26] Bowling, Lawrence Edward. "The Thematic Framework of *Romeo and Juliet*." *PMLA,* Vol. 64, No. 1 (Mar., 1949). 208.

Whether Bowling realizes it or not, his description of the undergirding structure of *Romeo and Juliet* is presciently digital. Bowling's argument, in both this excerpt as well as the majority of his article, is that Shakespeare's *Romeo and Juliet* is comprised of characters who view the world in a series of binaries – things like: good and evil; nobility and villainy; life and death; love and hate; Montague and Capulet. The ways in which Shakespeare organizes those binaries helps create, for both the characters themselves and the audience, the play's ultimately tragic complexities.

Juliet is perhaps the best example of a character who learns to distill complexity from simple binaries. Prior to meeting Romeo, Juliet assumed all Montagues are bad and all Capulets are good. But her love for Romeo forces her to reconsider what had once been an absolute binary of good and evil. This forced reassessment of her own binary beliefs is highlighted when the Nurse tells Juliet of Tybalt's death at the hands of Romeo. "O, that deceit should dwell / In such a gorgeous place," exclaims Juliet, unsure of how to rectify a binary she had already rebuilt once when she first discovered her love for a Montague (III.ii.83-84). As a result of the murder, Juliet is forced to reassess her binary view of Romeo and her binary view of her cousin Tybalt, wondering:

> But wherefore, villain, didst though kill my cousin?
> That villain cousin would have kill'd my husband.
>
> III.ii.100-101

The cousin who was, in her eyes, always good cannot remain that way once he tried to kill her husband. And the husband Juliet was easily able to flip from

one side of a binary (absolutely evil) to the other (absolutely good) no longer exists in such clearly defined categories. Instead, Romeo's slaying of Tybalt makes Romeo, for both the audience and for Juliet, a flawed hybrid of both good and evil, which turns him into a more complex character than he had been when either an inept young lover or a passionate and noble suitor.

Juliet's forced confrontation with complex binaries is one of many examples of Shakespeare creating complexity from the simple juxtaposition of opposites. Other often noted examples include: the Friar's consideration of the poisonous and medicinal effects residing in the same flower;[27] and Romeo's recontextualization of gold as poison and poison as something of value when dealing with the Mantua apothecary.[28] Shakespeare's forcing of these kinds of layers of complexity on top of extreme binaries foreshadows the current digital age – an age built atop a technology that generates complexity from binary code. While this dependence on binary in the digital age is rarely a focal point when using digital technologies, the most complex software programs are, at their core, and much like Shakespeare's *Romeo and Juliet*, deriving their complexity from binary opposites.

Although the digital tint of Bowling's binary argument is, considering the 1949 publication date of his article, likely an unintended aspect, the presence of a digital undertone in an analysis of *Romeo and Juliet* being published before digital technologies were widely available suggests that the mechanics of *Romeo and Juliet* and the mechanics of digital technologies

[27] "Within the infant rind of this weak flower / Poison hath residence, and medicine power" (III.ii.23-24).
[28] "There is thy gold – worse poison to men's souls, / Doing more murder in this loathsome world / Than these poor compounds that though mayst sell." (V.i.80-82)

have a relationship worth exploring. At the very least, the link between the core binary-ness of the two subjects encourages a logical follow-up question: Is the parallel structure between the basic operating principle of *Romeo and Juliet* and that of digital technologies a coincidence, or do the two subjects share overlap elsewhere?

To this question, Bowling gives yet another unintentionally digital response. That response appears in the title of his article: "The Thematic Framework of *Romeo and Juliet*." In Bowling's terms, the concept of a framework is serving as a description for the pre-established thematic structure on which *Romeo and Juliet* is composed. Its implication – an implication known to be true thanks to the work of Bowling and many other *Romeo and Juliet* scholars – is that the core thematic concepts of *Romeo and Juliet* are not Shakespeare's original constructs. Instead, *Romeo and Juliet*'s information architecture is derived from an established tradition of information dissemination that Bowling describes in the following way:

> During the Middle Ages, and to a lesser degree during the Renaissance, there was a marked tendency to view human beings and their actions in strict terms of wrong and right, black and white. To what extent this way of thinking was due to Judeo-Christian theology, it is difficult to say, but the tendency to think in terms of absolutes is certainly a Hebrew characteristic, evident in both the Talmud and the Bible. Opposed to this Hebraic tendency toward absolutes is the more or less characteristic Greek position that almost nothing is good

> or evil within itself but that particular elements are to be evaluated in terms of the relative situation. Again, it is difficult to say to what extent Renaissance ideals were influenced by Greek thought, but we do know that at the Renaissance there developed a revolt against the Hebraic view of life, in favor of the Hellenic. The meaning and significance of *Romeo and Juliet* may be more completely understood if we see the play as part of this greater movement toward a more relative and flexible view of human nature and human conduct.[29]

The Renaissance revolt against the Hebraic view of life via the appropriation of the Hellenic tradition connects with my second chapter and its discussion of how Homer was, and still is, a technology of information connection, storage, and dissemination. In this respect, the Romeo and Juliet story owes as much to Homer as it does to da Porto or Brooke since the framework Bowling alludes to goes back much farther than the traditionally cited *Romeo and Juliet* sources.

The connection to antiquity of the Romeo and Juliet legend underscores the ancientness of both the pre-established framework on which Shakespeare builds his story and his likely concept of "frameworks" as they are used in textual production. Frameworks provide Shakespeare, and other authors, already-existent thematic structures that pre-package the core information architecture and underlying functionality of a given informational object. While the fact that Shakespeare uses a pre-existing

[29] Bowling, 220.

framework is by no means new information, and has already been described earlier in this chapter when considering the story's history, what might be surprising is the existence of a nearly identical concept of pre-built frameworks in digital media which are the predominant creation tools for complex software applications.

Digital Frameworks, like literary frameworks, are pre-built software architectural constructs commonly leveraged by developers because they provide many core pieces for an application's most common architectural and functionality components. One advantage of using a framework to build complex pieces of software, as opposed to building software "from scratch," is that it lets developers avoid many of the more tedious basic development tasks such as creating login forms and user accounts, or adding common menu and interface elements such as the ability to save documents and resize objects. By applying the software concepts of a framework to a literary framework, the core plot elements of the Romeo and Juliet story that are replicated throughout its development could be viewed as the basic functionality anyone choosing to appropriate the framework will leverage in order to begin with a working though non-descript literary project. Those Romeo and Juliet framework elements include, among other common aspects, two rival socio-political entities, young members of these rivals falling in love, divisive conflict between the rivals which strains that love, and an ultimate reconciling (possibly, though not always, through death) of both the love and rivals.

But frameworks themselves are pieces of software. As pieces of software, they undergo the same kinds of development that any complex, dynamic piece of software undergoes, meaning that during their usage lifecycles they are regularly being developed and improved upon. That kind of development is also visible in the "Romeo and Juliet" framework. For example, in what would ultimately become Shakespeare's *Romeo and Juliet*, consider some of the following traceable framework elements: young lovers have a secret pseudo-wedding ceremony (*Ippolito e Leonora*); the murder and banishment events are added (Masuccio, *Il Novellino*); the "Romeo," "Juliet," and "Tybalt" names are incorporated into the story (Da Porto, *Giulietta y Romeo*); counselor-like figures are added for each lover (Bandello, Le Novellei, II, ix); the addition of the apothecary (Sevin); Juliet stabs herself with Romeo's dagger (Boaistuau, *Histoires tragiques*); an amplified Nurse character and the scene of Romeo's despair in the Friar's cell (Brooke, *Romeus and Juliet*); and, finally, Shakespeare's expansion of minor characters such as Mercuitio (addition) and Benvolio (expansion).[30]

With the framework in place, authors can then create their personalized texts by customizing the framework, adding the bits and pieces of functionality that make their particular implementation of the framework a unique application. In the case of the Romeo and Juliet story, such pieces of functionality include things like: the Romeo character getting arrested and the

[30] It's interesting to note here the possible use of what, in software development, are called "libraries." Libraries are pre-built pieces of functionality that can be added to software in order to easily incorporate additional functionality. The same corollary seems to exist in textual development. One example is the use of sleeping potions, which was popularized by Boccaccio's *Il Decamerone*. Another example would be lovers having a farewell in a tomb, which can be seen, among other places, in Ovid.

Juliet character having to publicly profess her secret love (*Ippolito e Leonora*); Romeo being the servant of the Friar-like character (Bandello); and the killing of Juliet's other suitor, Paris, in the tomb (Shakespeare). In addition, the particular stylistic and narrative tendencies of the texts and authors also appear to be personalizations of the core framework helping authors construct unique applications. For example, Shakespeare's use of blank verse and drama to present the story, though not unique to Shakespeare, suggest ways in which Shakespeare may be customizing the Romeo and Juliet framework in order to develop his own application.

The parallels between digital frameworks and the Romeo and Juliet framework establish a link between the constructive processes of texts and software. However, such connections are, by themselves, little more than surface-level similarities. The existence of core architectural structures in both literature and software is not new information, nor are framework-like tools limited to the literary and the digital. Most generative industries and disciplines rely, to some degree, on frameworks. For example, in music, Pachabel's Cannon in D might be considered a framework, and in architecture, Michelangelo's dome is a framework. My ability to recognize a digital concept of framework in literature (or a literary concept of legend and myth in software) is more a product of my having experience with software development than any discipline-altering interpretive truth hidden in the parallel itself.

If recognizing the existence of parallel development processes in literature and software only resulted in using digitally-bound terms to

describe literature, or textually-bound terms to describe software, then the parallels would probably be of little help. However, in the case of Shakespeare, the link between a literary concept of framework and a digital appropriation of the same concept has a usefulness that has the potential to transform the connection between the literary and the digital from one of jargon-based parallels into one with relevant social, historical, and cultural implications. Software frameworks – those digital tools used by developers to accelerate development I reference previously – do more than just help expedite the development of complex software applications. Frameworks are also critical to an application's security layer. They help developers build applications that are already embedded with comprehensive security mechanisms, meaning any new software built using frameworks are inherently more secure and less susceptible to data intrusions and corruptions.

Although concerns about information security are not usually related to literature, the structural parallels between digital frameworks and literary frameworks suggest that the same parallels might exist regarding security. Because frameworks are being used in both software and literary production to help authors more easily construct programs and texts, and because digital frameworks enhance digital security, are literary frameworks similarly helping enhance literary security?

By the time Shakespeare is writing in the late 16th and early 17th centuries, literary frameworks are a well-established toolset. In addition to

that toolset extending beyond *Romeo and Juliet* into Shakespeare's other plays (*King Lear*, *Macbeth*, *Hamlet*, etc.), the use of literary frameworks was both a common practice among Shakespeare's contemporaries as well as a standard practice for the majority of history's most highly regarded authors preceding Shakespeare, including, among many others, Chaucer, Dante, and Virgil. Understanding the commonality of Shakespeare's practice is critical for avoiding any assumptions that Shakespeare (or any other similarly-positioned author) was stealing, copying, plagiarizing, or otherwise maliciously appropriating the work of others. He was not. Instead, in Shakespeare's day, the practice of adopting popular motifs – what I have been referring to as the use of literary frameworks – was one of the primary ways in which authors demonstrated their talents. Good authors could leverage pre-existent texts or stories in unique ways that ultimately resulted in important structural milestones in the text's overall development.[31]

Shakespeare, intentionally or unintentionally, seems to have accomplished something slightly different. Instead of re-writing stories to demonstrate his talents and then enabling others to come after him and continue re-writing them, Shakespeare's versions often appear to take ownership of the stories themselves. The "Romeo and Juliet" story somehow

[31] Peter Stallybrass makes note of the difference in the ways imitation and intellectual property are perceived and understood today versus how they would have been recognized prior to the extreme cultural regulating of concepts of intellectual property when he writes: "Learning requires imitation and inspiration, which today are marginalized by a concept of originality that produces as its inevitable double the specter or plagiarism, a specter rooted in the fear that we might have more to learn from others than from ourselves." ("Against Thinking," 1583). In this particular article, issues of intellectual property are only briefly taken up by Stallybrass, but it points to larger questions about how intellectual property was understood prior to the Enlightenment. Joseph Loewenstein devotes more time to discussing concepts of copyright in Shakespeare's time in both *The Author's Due* and *Ben Jonson and Possessive Authorship.*

turns into not Brooke, Bandello, or da Porto's "Romeo and Juliet," but *William Shakespeare's Romeo and Juliet*. How do his texts do this? Perhaps Shakespeare discovered security vulnerabilities within certain literary frameworks which he ultimately exploited to take ownership of them.

Jill Levenson describes Shakespeare's transformation of the Romeo and Juliet framework into *Shakespeare's Romeo and Juliet* as being a product of the poet's stylistic and artistic talents. She suggests that, prior to Shakespeare, "the Romeo and Juliet fable held its unchanging course as a simple and sentimental narrative. Because the *novellieri* exploited only its plot and pathos, it continued to be, until the end of the sixteenth century, a lachrymose description of fervent passion."[32] Shakespeare's version, according to Levenson, changes all of that. "In its brilliant fusion of several artistic conventions," Levenson argues:

> *Romeo and Juliet* seems to me a drama that invites analysis. Its well-known sequence plays itself out through diverse sets of stylistic techniques in constantly changing relationships. Wherever one looks, this kaleidoscopic effect overturns stereotypes and tests the flexibility of aesthetic customs. Together, the familiar love story and Shakespeare's experimental style engage not only emotion but also wit. Like Hamlet's love poem to Ophelia, *Romeo and Juliet* is an

[32] Levenson, Jill L. "Romeo and Juliet before Shakespeare." *Studies in Philology*, Vol. 81, No. 3 (Summer, 1984). 328.

> intellectual's posy; it anatomized the very conventions in which it originated.[33]

While Levenson notes a clear distinction between the quality (i.e. literary success) of Shakespeare's *Romeo and Juliet* versus other Romeo and Juliet iterations, her argument for the cause of that success is based on opinion-based artistic merit. But artistic merit is determined by aesthetic judgments, and using such judgments to explain why the Romeo and Juliet framework has become so closely associated with Shakespeare seems hard to quantify and difficult to justify. Aside from the inherent personal and cultural biases influencing conceptions of artistic merit, framing the artistic merits of Shakespeare as being responsible for his success implies that previous versions of the Romeo and Juliet story had little or no artistic merit worth mentioning. However, the ability of the story itself to reach Shakespeare from Italy by way of France over the course of hundreds of years would seem to contradict Levenson's approach. Earlier versions of the Romeo and Juliet framework were pleasurable enough to contemporary readers that "Shakespeare's fusion of brilliant artistic conventions," while worthy of praise, might not be the only aspect that convinced people to agree the Romeo and Juliet story was worth preserving. By the time Shakespeare wrote his *Romeo and Juliet*, the framework had already successfully been preserved, in a clearly definable form, for generations, and the foundations of the overall framework had been in place for centuries.

[33] Ibid. 347.

By considering *Romeo and Juliet* from a perspective less interested in Shakespeare's artistic prowess and more interested in the story's implementation as a literary framework, Shakespeare's success with the Romeo and Juliet framework could be understood using logic that is less reliant on aesthetic judgments and instead based in questions of data security. Soon after Shakespeare develops and deploys the framework to suit his vision of the Romeo and Juliet legend, development of the framework itself, if titles and popular culture are to be believed, reaches a definitive stopping point. Any lasting development done to the framework after Shakespeare is done not to *Romeo and Juliet,* but to *Shakespeare's Romeo and Juliet*. This change in the way people perceived the popular story suggests that whatever vulnerability Shakespeare exploited in the Romeo and Juliet framework, the effect seems to have been that the story changed from being a common cultural story into William Shakespeare's story.

A good way to approach the possibility that Shakespeare is implementing a more secure framework is to begin by recognizing that development of the Romeo and Juliet framework did not actually stop after Shakespeare. One notable example, among many, is *West Side Story*. Despite being set in New York City's Upper West Side neighborhood instead of "fair Verona," despite the warring factions being ethnicities instead of families, despite the characters having different names, despite the plotlines being substantially different, and despite the large alterations to staging and presentation mechanisms characteristic of Broadway musicals when compared with Shakespearean theatre, *West Side Story* is not commonly

perceived as an iteration of the Romeo and Juliet framework. *West Side Story* is an iteration of *Shakespeare's Romeo and Juliet*.[34]

The existence of an extremely successful Broadway musical (and movie) created 350 years after Shakespeare's *Romeo and Juliet* and with few linguistic ties to the Shakespearean version of a well-established framework nevertheless being understood as a derivation of Shakespeare (and not a derivation of the Romeo and Juliet legend) could be a clue of Shakespeare's successful hacking. If Shakespeare was exploiting a vulnerability in the Romeo and Juliet framework, the data vulnerability would seem to be whatever vulnerability was allowing the Romeo and Juliet framework to be constantly re-appropriated and re-authored by whichever author(s) next implemented it. In contrast, Shakespeare's exploit appears to have secured the framework by creating a standard version. Any deviations from that version, while acceptable (*West Side Story* being one of many notable and not-so-notable examples[35]), all have a common point of popularly-accepted ancestry not in da Porto and not in Bondello and not in Brooke, but in William Shakespeare.

The transition of the Romeo and Juliet framework into Shakespeare's *Romeo and Juliet* could be a clue to a shift in literary transmission structures

[34] A helpful description of the eight-year generative process between Arthur Laurents, Leonard Bernstein, Stephen Sondheim, and Jerome Robbins that eventually resulted in "set[ting] Shakespeare's *Romeo and Juliet* in contemporary New York City" can be found in Brian Eugenio Herrera's "Compiling *West Side Story*'s Parahistories, 1949-2009," specifically pages 234-238.
[35] Other examples of the Romeo and Juliet story being adapted range from an animated movie about garden gnomes called *Gnomeo and Juliet*, to a Royal Shakespeare Company production of *Romeo and Juliet* performed on Twitter called *Such Tweet Sorrow*.

that begins shortly after Shakespeare's time. In general, before William Shakespeare, authorship was secondary to a story. Stories were transmitted from one generation to the next, while authorship of individual versions often changed. Foucault notes this earlier form of textual transmission by explaining that:

> The same types of texts have not always required authors; there was a time when those texts which we now call "literary" (stories, folk tales, epics, and tragedies) were accepted, circulated, and valorized without any question about the identity of their author.[36]

However, sometime not too long after Shakespeare, authorship began to transition to its current role as a primary delineator of a text's information transmission protocols. Foucault explains how, in the seventeenth century:

> "Literary" discourse was acceptable only if it carried an author's name; every text of poetry or fiction was obliged to state its author and the date, place, and circumstance of its writing. The meaning and value attributed to the text depended on this information.[37]

This evolution in the ways in which texts are packaged with their authors might, in retrospect, be difficult to recognize because our modern inclination is to associate an author with a story, even if that author existed prior to Shakespeare. However, consider the example of Homer from my previous chapter. In England, prior to Shakespeare, people had little concern for

[36] Foucault. 123.
[37] Ibid. 126.

Homer, yet they had a long-held cultural understanding of the Troy Stories, which were stories derived in and around Homer-centric myths. One notable example of the lacking concern for Homer despite the prominence of the Troy Stories comes from Shakespeare himself and the drama *Troilus and Cressida*. A previous incarnation of the same tale was famously told by Chaucer more than a century earlier when he wrote *Troilus and Criseyede*. However, the English interest in the Homeric Question – that question of "Who was Homer?" – begins to appear in the wake of Shakespeare. Is it simple coincidence that a prioritizing of Homer began so soon after Shakespeare's ascension on the Elizabethan stage, or is Shakespeare hacking literature? Is Shakespeare exploiting a vulnerability in the technology of narrative transmission, and, as a result, forcing the cultural maintainers of that technology to implement protective processes to prevent the same exploit from being used again?

These kinds of questions do not have simple "yes" or "no" answers. Additionally, the influence of Shakespeare is, at least in part (or perhaps in whole), a likely byproduct of readers embedding a certain authority in his name rather than in Shakespeare actually establishing that authority himself. That is to say "Shakespeare is Shakespeare" in the same way "Homer is Homer" – "he," as discussed in this chapter, seems to be more of an idea than an actual person. Regardless of how the Shakespeare idea has become so influential, Shakespeare was only a contributor to, not solely responsible for the evolution of literary security mechanisms like copyright, intellectual property law, and concepts of authorship. Similarly, Robert Morris and the

Morris Worm were not solely responsible for all of the mechanisms of Internet security. But, intentional or not, Morris certainly impacted the development of Internet security technologies, just as Shakespeare has influenced literary security mechanisms of authorship and ownership. And because of Shakespeare's success in hacking literature, his work, like Morris's work, offers insights into the ways in which data security protocols develop within information technologies.

In the case of Robert Morris, when the Morris Worm infiltrated the Internet and crashed a large portion of its computers, maintainers of the Internet responded by creating a technology they called a firewall. A firewall serves as a protective barrier for computers and computer networks by filtering all incoming and outgoing traffic through a pre-defined set of qualifications. If the data hitting the firewall does not meet the qualifications of what the firewall believes are non-malicious pieces of information, the firewall is able to prevent the data from reaching its intended recipients. This security process provides an effective, long-term barrier to data corruption because the process itself is not pre-determined by the data. Instead, the filtering process adapts to the kinds of data being transmitted.

In the case of literary data transmission, a similar security processing mechanism seems to have evolved in the form of an author's name and an author's version of a story becoming affixed to common classical tales. Foucault describes this phenomenon by explaining that, "The function of an author is to characterize the existence, circulation, and operation of certain

discourses within a society."[38] The author's name, according to Foucault, enables a pre-defined set of qualifications for the data within a story, and no matter how that data is corrupted by later versions, the named-story remains stable and protected. In the example of *Romeo and Juliet*, prior to Shakespeare, the data of the story was highly unstable. The result was the rapid redevelopment and redeployment of the story in a variety of mediums with a variety of differing storylines, while no single version served as a sort of baseline or Urtext of common information. After Shakespeare, deployments of the Romeo and Juliet framework remain common, but those deployments are often traced (regardless of the actual origin) to a single, "standard" version of the story, preventing the data from being corrupted and changed with each iteration. Instead, new iterations of the story regularly return to Shakespeare's version.

A powerful example of the Shakespeare name acting as a protective firewall on the Romeo and Juliet story is, once again, *West Side Story*. Despite the immense popularity of the Broadway musical, and despite the even greater popularity of the subsequent movie – it won 10 Academy Awards – *West Side Story* has so far done little, if anything, to permanently alter the Romeo and Juliet story for future authors. Instead, a few years later, Franco Zeffirelli's popular 1968 version of *Romeo and Juliet* returns to a Shakespearian rendition of the storyline, including period costumes and settings. Where was the influence of *West Side Story*, one of history's most successful implementations of the Romeo and Juliet framework? Three

[38] Foucault. 123.

decades after Zeffirelli, the next prominent iteration of the Romeo and Juliet framework, Baz Luhrmann's movie adaptation, is titled not just "Romeo and Juliet" (or "Tony and Maria"), but "*William Shakespeare's Romeo + Juliet*," which, despite its many visual and conceptual departures from Shakespeare's play as described at the beginning of this chapter, seems to be leveraging the Shakespeare name as a protective tool and a marketing strategy. By directly appropriating the security mechanism of the Shakespeare name, Luhrmann can define his implementation of the framework not as a new piece of technology that audiences must judge in comparison to Shakespeare's canonical iteration of the framework, but instead as a deployment of Shakespeare's original text. In this regard, the name of Shakespeare seems more important to Luhrmann's movie than Shakespeare's language.

In what could be a subtle nod toward the technological vulnerability within literature that Shakespeare would exploit, Juliet's simple question becomes an opportunity for exploration about how information is packaged and transmitted in literary texts when she famously asks: "What's in a name?" According to Shakespeare, names have the unmatched power to create protective metaphysical walls around informational blocks. In so doing, they secure the storage and transmission of information in a way that, as the ever-changing history of the Romeo and Juliet framework prior to Shakespeare suggests, brings stability and unity to complex ideas.

The stability and security created by attaching names to ideas and concepts is not without its price, as Shakespeare reminds his audiences. The

divisions represented by the Montague and Capulet names are what cause the deaths of young Romeo and Juliet. But, at the same time, names seem to provide order in both their world and ours. Romeo reminds audiences of this distinction when, upon learning of his banishment, he dejectedly proclaims: "There is no world without Verona walls" (III.iii.17). Romeo is right. Without the artificial boundaries created by mankind – things like city names, country names, family names, and even the names of people – the world is a difficult place to navigate. But Shakespeare, by hacking literature and having his name attached to so many common stories, has helped make the literary storage and transmission technologies used to navigate that world safer, more stable, and more secure.

Chapter 4: This Particular World Wide Web

A similarity between George Eliot's early 19th century, fictional British town of Middlemarch and one of the world's most prominent nuclear research laboratories, the European Organization for Nuclear Research (commonly known as CERN), appears in the following description of CERN written, in 1989, by one of its fellows, Tim Berners-Lee. Berners-Lee a CERN computer scientist from 1984-1994, described the operational culture of the multi-thousand-person research facility on the border of France and Switzerland by explaining that:

> The actual observed working structure of [CERN] is a multiply connected "web" whose interconnections evolve with time. In this environment, a new person arriving, or someone taking on a new task, is normally given a few hints as to who would be useful people to talk to. Information about what facilities exist and how to find out about them travels in the corridor gossip and occasional newsletters, and the details about what is required to be done spread in a similar way. All things considered, the result is remarkably successful, despite occasional misunderstandings and duplicated effort.[1]

This description of CERN parallels the experience of Middlemarch's Tertius Lydgate. As the town's newly-arrived doctor, much of his first appearances in the book describe his learning about the town's intricate relationship

[1] Berners-Lee, Tim. "Information Management: A Proposal." May 1990. Web. 13 Jul. 2012. http://www.w3.org/History/1989/proposal.html.

networks. That knowledge comes not through any easily accessible archival repositories, but by attending social events and parties – the Middlemarch-equivalent of "corridor gossip and occasional newsletters."[2]

Though demonstrated in entirely different contexts and circumstances, in their descriptions of the ways people share information between one another, both Eliot and Berners-Lee are presenting the natural flow of information within complex community systems. They are both, very famously, describing connective informational webs.

The development of Tim Berners-Lee's technological web began as a project to provide CERN with a better information storage and dissemination technology. The above quote describing CERN provides the introductory context of the then-current state of affairs at the research lab, and it begins a proposal in which Berners-Lee addresses what he believed was one of CERN's biggest problems: losing information. Further explaining the problem, Berners-Lee describes how the high turnover rate of CERN staff combined with the large amounts of research CERN scientists were performing was causing "technical details of past projects [to be] sometimes lost forever, or only recovered after a detective investigation in an emergency. Often, the information has been recorded, it just cannot be found."[3] While Berners-Lee was not addressing an information access problem unique to CERN, Berners-Lee had the technical background and the physical resources to help him develop a digital solution.

---

[2] The first time Lydgate is introduced in the book comes at the end of Chapter 10. He is leaving one of Dorothea and Casaubon's engagement parties and is reflecting on the information about the town and its people that he gathered during the course of the event.
[3] Berners-Lee, "Information Management: A Proposal."

After graduating with a degree in physics from Oxford in 1976, Berners-Lee worked for a UK telecom equipment manufacturer where he focused on building distributed transaction systems. Those distributed transaction systems would inform his work during his first, briefer stint at CERN, when he spent half a year in 1980 working at a CERN lab as a consultant. Berners-Lee describes the organizational complexity of the staff network he encountered during the time, explaining that:

> Of the (approximately) 10,000 people in the CERN phonebook, only around 3,000 were actually salaried CERN staff. Many of the others have a desk at CERN and also one in their home institute, and they visit every now and again. These people report to and are paid by their home institute. Whilst at CERN, collaboration occurs by peer-peer agreement and academic mutual respect. This makes it impossible for anyone to make (or enforce) global decisions on types of hardware and software to be used. People bring with them machines and customs from their home universities. You just have to do your best to accommodate them. This had already led to a lot of work on mail and file interchange between different types of computer system and different networks. [*sic*][4]

As Berners-Lee describes it, by 1980 CERN was so complex a network that enforcing a standardized organization system across the community was

[4] Berners-Lee, Tim. "A Brief History of the Web." 1993. Web. 14 Jul. 2012. http://www.w3.org/DesignIssues/TimBook-old/History.html.

impossible. Despite that impossibility, Berners-Lee felt he could create his own organization system to at least help him keep track of information, and he could make the system open so other people could use it as well. The result of that effort was "Enquire," a computer program for his own personal use designed to:

> Note down all the dependencies between people, programs, and hardware. It was a bit like a card index, but with links between cards. It ran on a plain 24x80 terminal. The links had meanings like "A is part of B", or "A made B". The idea was that when one had put in everything one knew, one could trace through implications, answering questions such as "If I remove this module, what stops working?" [5]

Berners-Lee never publicly published Enquire, but he personally used it often, and he even had moderate interest from others at CERN who leveraged it as a useful way for keeping organized.

In 1984, Berners-Lee received a fellowship and returned to CERN in order to "work on distributed real-time systems for scientific data acquisition and system control."[6] Developing these systems put him in a position of working with a highly dispersed and mobile group of people who often worked outside of CERN. In order to organize and disseminate up-to-date information about projects, Berners-Lee leveraged Enquire, and by doing so, he ultimately exposed its primary conceptual flaw: Because Enquire used

[5] Ibid.
[6] "Longer Biography." Web. 14 Jul. 2012. http://www.w3.org/People/Berners-Lee/Longer.html.

proprietary protocols, anytime the work of Berners-Lee's project intersected with similar work being done by other CERN researchers, incorporating the other people's work into Enquire forced Berners-Lee to spend large amounts of time organizing information. The effort required to merge information had Berners-Lee longing for a more efficient process. "There was clearly a need for something like *Enquire*," Berners-Lee writes, "but accessable to everyone. I wanted it to scale so that if two people started to use it independently, and later started to work together, they could start linking together their information without making any other changes [*sic*]."[7] This concept of digitally linking the individual documentation sets of two distinct informational repositories became the foundation for what would become the World Wide Web.

Berners-Lee quickly realized his own needs for such a system were only a microcosm of a much larger CERN problem. In an organization with thousands of researchers and hundreds of research projects, Berners-Lee knew that much of the work being done at CERN intersected within departments, operational groups, and projects. He reasoned if his project could benefit from the kind of improved information sharing offered by a digitally interlinking network of separately-maintained informational repositories, surely others could as well.

By chance, the timing at CERN was right for such an idea to take root. In 1989, CERN had recently completed its Large Electron-Positron Collider, and plans for the development of the Large Hadron Collider were still a few

[7] Berners-Lee. "Brief History."

years away, leaving the research lab with a sort of development "downtime." After numerous largely ignored proposals, Berners-Lee's directors finally gave him permission to work on an idea that, at the time, he referred to as a "mesh."

The fact that Berners-Lee envisioned a “mesh” as the undergirding connectivity metaphor for the World Wide Web seems to go against popular perception that the “web” metaphor comes from the concept of a spider’s web.[8] Instead, Berners-Lee’s “web” metaphor is a comparison to mesh fabric. Recognizing this use of the webbed fabric metaphor provides an important clue to the potential connection I suggest to introduce this chapter – the connection between CERN and George Eliot’s *Middlemarch*. George Eliot made the same fabric-as-web association over a century earlier when, at the beginning of Chapter XV of *Middlemarch*, her narrator busily and famously explains, “I at least have so much to do in unraveling certain human lots, and seeing how they were woven and interwoven, that all the light I can command must be concentrated on this particular web, and not dispersed over that tempting range of relevancies called the universe” (1951-1953).[9]

In her book *Darwin’s Plots: Evolutionary Narrative in Darwin, George Eliot and Nineteenth-Century Fiction*, Gillian Beer explains how

[8] One prominent example of the more commonly assumed spider web metaphor is the title given to computer programs that “crawl” websites, such as those deployed by search engines in order to catalogue information: They are generally called “spiders.”

[9] Again, keeping with the spirit of this project, for my edition of *Middlemarch*, I chose a free and openly available edition of the text available for download on Amazon.com’s Kindel e-reader. All citations will rely on Kindle location numbers from the following version: Eliot, George. *Middlemarch*. New York and Boston: H. M. Caldwell Company Publishers.

George Eliot's connotation of a web would have been different from a late 20th century interpretation of the word, explaining that:

> For us now, the spider's web is probably the predominant association of the word 'web'. But for Victorian people, woven fabric seems to have been the predominant reference. Web imagery is to be found everywhere in Victorian writing. It is as common among scientists and philosophers as it is among poets and novelists.[10]

First published in 1983, Beer's "for us now" assertion is, three short decades later, already dated. While the spider web may have been the predominant association of the word "web" in 1983 (and, presumably, the same was true around the time of the conceptualization of the World Wide Web in 1989), the word "web" now, as a result of Tim Berners-Lee, likely conjures the very different image of Berners-Lee's Web. Since that web – the World Wide Web – has its structural roots in the concept of a meshed fabric, the fact that George Eliot's *Middlemarch* web should also be imagined as a fabric-based web suggests a connection between the two technologies. This potential connection has lead me to wonder what the connective structure of the World Wide Web reveals about George Eliot's *Middlemarch*, and what *Middlemarch* can reveal about the World Wide Web.

***

[10] Beer, Gillian. *Darwin's Plots: Evolutionary Narrative in Darwin, George Eliot and Nineteenth-Century Fiction*. Cambridge: Cambridge University Press, 2009. 156.

As my enquiries suggest, structural and thematic revelations about *Middlemarch*, a story famously described by its narrator as a connective web, become possible by understanding the structuring of *Middlemarch* to be in some way related to the structure of the World Wide Web. My analysis of *Middlemarch* begins, as a result, with an analysis of the Web, and I will begin that analysis at the point of most confusion: the Internet.

Most people use the terms "Internet" and "World Wide Web" interchangeably. Doing so, however, underscores a fundamental misunderstanding of what each one is, how each one operates, and what purposes each one best serves. The Internet is a network of interconnected digital networks that communicate with one-another through a series of digital communications protocols known as the Internet Protocol Suite. More simply, it can be thought of as a collection of digital networks connected to each other by the same underlying technological principles. Tim Berners-Lee's World Wide Web is one of the networks connected via the Internet, but it is one of many. What this means, from a general consumer perspective, is that the Internet itself is the near ubiquitous network enabling the connections between everything from our cell phones to our ATM machines to our GPS satellites. The World Wide Web is a network that operates within the Internet but is not, itself, the Internet.

I realize such distinctions might seem pedantic, but they are important for understanding the World Wide Web's purpose as a technology of information storage and dissemination. Making a call with a cell phone means using the Internet; sending a text message with a cell phone means using the

Internet; and browsing a webpage from a cell phone means using the Internet. But calling and texting are not examples of using the World Wide Web, whereas using a web browser to browse a webpage is. The distinction at issue in these differences emphasizes the Web's significance as not just a connective technology, but a technology that enables people to connect with very specific types of information in very specific ways.

The World Wide Web Consortium (W3C), an international organization co-founded and led by Berners-Lee that develops web standards, calls the World Wide Web "an information space in which the items of interest, referred to as resources, are identified by global identifiers called Uniform Resource Identifiers (URI)."[11] Based on the W3C's explanation, as well as the Web's integration within the Internet, users of the World Wide Web should understand it not as the Internet itself, but as one network within the Internet that provides them with a globally accessible information space in which they can connect to different resources through a protocol of unique URI identifiers.[12]

The current structure of the World Wide Web appears to be a significantly scaled version Berners-Lee's original vision. In his proposal for CERN, he argued for the necessity of individually created and maintained informational spaces capable of being accessed remotely and capable of connecting with any other individually maintained and remotely accessible

[11] "Architecture of the World Wide Web, Volume One." *W3C*. 15 Dec. 2004. Web. 17 Jul. 2012. http://www.w3.org/TR/webarch/. Section 1.
[12] URIs are more commonly recognized by the terms "URL" or "domain name." The alternate terms are not synonymous, but the distinction is not relevant to this chapter's discussion. A technical explanation is available at http://www.w3.org/TR/uri-clarification/.

informational space built using the same protocols and specifications. Websites are the individually maintained, remotely accessible informational spaces Berners-Lee describes, they connect with one-another through hyperlinks that rely on a system of unique identifiers, and they are all created using standardized specifications.

Berners-Lee's original vision was one intended for CERN. It was meant to help CERN researchers more easily organize and connect with the massive quantities of otherwise inaccessible information being produced by fellow CERN researchers. But the widespread adoption of the World Wide Web as a daily tool in the lives of billions of people around the world suggests that the problem Berners-Lee was attempting to solve when he invented the World Wide Web for CERN was also a problem for the world beyond CERN.

The problem itself – what Berners-Lee describes in his original proposal for the World Wide Web as "losing information" – was, in 1989, not a new problem. As my earlier chapters discuss, as long as there has been recorded history (and, indeed, recorded history itself suggests the problem began earlier), people have been trying to overcome the natural loss of information. This is the same struggle which, as my first chapter explains, Freud attributes to the "death drive" and Derrida names "archive fever." Martha Nell Smith, a digital humanities pioneer, explains the common practice of relying on new technologies to combat archive fever when she notes:

> That scholars and teachers have been thoughtfully applying technologies in their research and instructional praxes is

> nothing new: scholars and teachers have been doing this long before the Internet, long before the computer, long before the typewriter, long before the pencil, long before A.D. and B.C. marked the accounting of time. Our knowledge building would be important whether we had digital tools or not.[13]

Smith's point is a reminder of what I have previously noted at the beginning of this project and what most digital champions and critics already realize: that digital tools are only the most recent example of the latest technologies being applied to the task of knowledge storage and dissemination. However, Smith makes her point in conjunction with a reality that often seems to receive less attention. She reminds us that, "The fanciest computational software can do nothing, nothing interesting at all, unless directed and engaged by the most important software of all – that proffered by the human touch, by, in other words, *you/us/me*" [emphasis in the original].[14] Smith's argument is that no matter the unprecedented power of the technology, the technology itself does not produce, it only retains. The user of the technology is the person responsible for leveraging the resource in productive ways.

In the case of the World Wide Web, because it is a technology created to improve access to constantly changing information, it is a technology created because a previous technology serving the same purpose was unable to adequately serve the same needs. In Berners-Lee's proposal for the World

---

[13] Smith, Martha Nell. "The Human Touch – Software of the Highest Order: Revisiting Editing as Interpretation," presented at Digital Textual Studies: Past, Present and Future, Texas A&M University Department of English, October 2006. 3

[14] Ibid.

Wide Web, he directly addresses the problems of one such previously developed and inadequate technology, explaining that:

> If a CERN experiment were a static once-only development, all the information could be written in a big book. As it is, CERN is constantly changing as new ideas are produced, as new technology becomes available, and in order to get around unforeseen technical problems. When a change is necessary, it normally affects only a small part of the organisation. A local reason arises for changing a part of the experiment or detector. At this point, one has to dig around to find out what other parts and people will be affected. Keeping a book up to date becomes impractical, and the structure of the book needs to be constantly revised. [*sic*][15]

The technology unable to serve CERN's connectivity and preservation needs, according to Berners-Lee, is the technology of the book. He concludes that the reason the book is insufficient for CERN is because CERN is, when compared to the rest of the world, more informationally advanced. He makes this argument when he writes:

> The problems of information loss may be particularly acute at CERN, but in this case (as in certain others), CERN is a model in miniature of the rest of world in a few years time. CERN meets now some problems which the rest of the world will have to face soon. In 10 years, there may be many commercial solutions to

[15] Berners-Lee, "Information Management."

> the problems above, while today we need something to allow us to continue.[16]

By recognizing the need for a more dynamic information storage and dissemination tool, Berners-Lee was solving a problem both he and numerous other researchers struggled with. As most researchers interested in publishing their work can attest, the technology of the book is not capable of fast and easy regular updates, but the shared protocols of the World Wide Web have helped overcome part of that limitation. Additionally, Berners-Lee's projection about the rest of the world seems to have been accurate. The rapid adoption of the World Wide Web outside CERN within 10 years confirms that the world was also in need of better information technologies.

Since the rapid adoption of the Web by the world outside of CERN suggests the general population also wanted a rapid and dynamic connective information space, to assume that technologies prior to the World Wide Web were not being used to accomplish – or, at least, to attempt to accomplish – the same goals of efficient information storage and dissemination would seem myopic. As the remarks noted above by Martha Nell Smith explain, no matter the technology, the most important software is the human software operating it. And that software, the human software that operates web pages, is the same software that operates books. It also operates other historically impactful cultural and archival technology-driven information spaces like drama, epic, and music. Surely that software – the human software – still

[16] Ibid.

desired more quickly updating, globally accessible information spaces long before Berners-Lee developed the Web.

Before considering how pre-digital technologies like books were still attempting to connect users with a form of intertwined and continually updating information space, I can help my discussion by explaining that the World Wide Web was not the first, nor is it the only digital information space to leverage the digital connectivity of the Internet. A brief investigation of the Internet's history reveals how leveraging its potential as a rapidly updating, interlinked, communal information space was a goal of many of its earliest pioneers. In 1962, for example, J.C.R. Licklider, one of the originators of the concept of "packet switching" (the core technological information transmission component of the Internet) proposed the concept of a "Galactic Network" in which "he envisioned a globally interconnected set of computers through which everyone could quickly access data and programs from any site."[17] Shortly thereafter, in 1963, another Internet pioneer named Ted Nelson coined the terms "hypertext" and "hypermedia" as he envisioned a connective information space similar to Berners-Lee's World Wide Web concept. Nelson called his concept Project Xanadu(R),[18] and he continues to vehemently defend it as a preferred alternative to the World Wide Web,

[17] Leiner, Barry M. et al. "Brief History of the Internet." *Internet Society*. Web. 18 Jul. 2012. http://www.internetsociety.org/internet/internet-51/history-internet/brief-history-internet.
[18] The name of the project comes from Coleridge's "Kubla Khan." Nelson requests that all first uses of the project name be given proper copyright attribution when he writes: "'Xanadu' is a registered trademark which I maintain at considerable cost, and I ask all parties to respect this by using the "®" or "(R)" symbol for the first use of the trademark 'Xanadu' in each document" (http://hyperland.com/TBLpage).

claiming its structure can correct what Nelson believes to be many of the Web's shortcomings that resulted from Berners-Lee's over simplifications.

Recognizing that the goal of using the Internet as a connective information space was not an idea that originated with CERN or Tim Berners-Lee underscores the consistency of the human software. By the time digital computers were barely common enough for some of the world's most resource-rich institutions to afford them, their earliest users had already begun envisioning the ability of such tools to establish a connective, dynamic, communally accessible information space. Such lofty goals for so new a technology suggests a desire to accomplish the same work long before the World Wide Web made it common. As a result, we should expect to find evidence of similar work being attempted in one of the most popular mass-media public informational storage and transmission technologies preceding the World Wide Web: the novel.[19]

My discussion turns toward the novel with the realization that novels were not the only popular form of connective print media serving as an information space prior to the World Wide Web. One notable example, among many, is newspapers. Newspapers not only serve the purpose of connecting people with a shared set of cultural information (i.e. the "news" as it represents current cultural and political events and interests), the physical nature of newspapers enable them to be retained and accessed at a later date

[19] Other technologies for information transmission did come between the novel and the Web – things like the telegraph, telephone, and television. The distinction I'm making here is that information storage is important (i.e. television and radio do not provide any new mechanisms for information storage).

in order to retrieve archival information. A novel, in contrast, is a printed and therefore standardized text that, when compared with newspapers, usually has a more limited scope of content. Despite their relatively limited content, the ability of novels to be read across many cultures and communities, particularly as the costs of printing began to decrease significantly in the 1800's, helped make novels highly successful communal information spaces through which readers (i.e. "users") can remotely connect with information. In this respect, like the World Wide Web, novels are a form of technology that provide remote access to shared information spaces.

While novels can accomplish the basic work of establishing remote connections to shared information spaces for which the Web receives so much praise, the information within novels is not as easily updated, nor do novels so directly connect users with external informational resources. Because of these shortcomings, the novel, as a technology, is limited in its ability to prevent information loss. But condemning the technology of printed media in general, and novels in particular, for their inability to prevent information loss, as Berners-Lee does in his proposal for the World Wide Web, overlooks the driving force behind any technological advancement in communication media: the human software. In this case, praising the technology of the World Wide Web for overcoming limitations of information loss inherent in print media suggests that preserving information was not a fundamental purpose driving the development of printed texts. It was. As Derrida emphasizes in his concept of archive fever, the human software is intent upon preserving and having access to information. Print media, which includes novels, is a

technology of cross-generational information preservation, making print media, like the World Wide Web, a human technology for preventing information loss.[20]

Since people have continuously sought improved communication technologies for information storage and dissemination, and novels are a form of communication technology, then perhaps one metric by which users can assess the successfulness (or shortcomings) of a novel is to consider how well a given novel prevents information loss. Phrased more casually, I wonder if a novel largely regarded as "great" is achieving its greatness, at least in part, because of its ability to more successfully achieve the technological goal of preserving and communicating shared cultural information. Are great novels more than just well-crafted narratives? Are great novels, like web pages, successful shared information spaces?

The question itself admittedly carries a burden of impracticality because, not only does it rely on a value judgment for what constitutes “great,” it also likely frames greatness in respect to current assessments of a novel’s purpose and structure. What makes a novel great? And what makes a novel great in the opinion of one reader at one time in history might not make it great in the opinion of another reader at another time in history. But setting aside value judgments and historical moments, by returning to George Eliot's *Middlemarch* we can consider a text that, regardless of greatness, is certainly highly regarded for its ability to convey a sense of realism. "*Middlemarch* is

[20] Similarly, to presume the World Wide Web is sufficient for preventing information loss would be to suggest people now have access to all the information they need at all times and that the Web will never be supplanted by a future information storage technology. Such a presumption, as history shows in its constant push for technological advancement, is certainly false.

not merely the Everest of English realism," Calvin Bedient asserts, "it is the epitome of realism itself… Beside it the rest of English fiction appears – fictitious."[21] A book so highly regarded for its successful realism would seem a likely candidate for being adept at preserving and conveying shared cultural information.

Because *Middlemarch* is so highly regarded for being a text of realism – for being a text that does the work of portraying accurate representations of the human condition within authentic surroundings – I propose it as novel well-positioned to initiate a discussion of how novels are technologies of shared (i.e. "real" as it relates to the accuracy of depicting human events and interactions) information spaces. The realism of the novel helps to situate its content within both a fictitiousness inherent to novels and a reality that suggests the content, while imagined, could be factual. From that bridge between the imagined and the real, I hope less realistic fictional texts will be easier to accept as information spaces of the "real." In addition, in *Middlemarch*, Eliot herself might be showing exactly how readers can make that connection.

In order to understand how Eliot may be empowering her chosen technology – the technology of the novel – with the ability to operate as a shared information space, I will begin with what might be perceived as the near-heretical task of severing the book, at least temporarily, from its

[21] Bedient, Calvin. "*Middlemarch*: Touching Down." *The Hudson Review*, Vol. 22, No. 1 (Spring, 1969). 70-71.

beginning and ending. I call such work "near-heretical" because a large swath of the critical discussion surrounding *Middlemarch* attends, at least in some ways, to two significant textual moments, one found in the opening Prelude and one in the closing Finale. Note that I do not mean to completely disregard the beginning and ending of *Middlemarch*, but I mean to propose a reason those two moments might need additional consideration beyond the interpretive possibilities they offer.

The first of the two moments I want to reconsider is introduced in the very first sentence of *Middlemarch*, when George Eliot wonders about Saint Theresa. A few paragraphs later in her Prelude, Eliot's narrator muses on the "many Theresas [who] have been born who found for themselves no epic life wherein there was a constant unfolding of far-resonant action" (10-11). Much has been written that presumes *Middlemarch*'s seeming-heroine, Dorothea, is an implied example of one such "many Theresas" who would go on to have non-epic lives.[22] That assumption seems to be verified by the second oft-noted textual moment in *Middlemarch*'s finale when Eliot changes "many Theresas" into "many Dorotheas," and she concludes the book with a suggestion that "things are not so ill with you and me as they might have been, is half owing to the number who lived faithfully a hidden life, and rest in unvisited tombs" (11987-11988).

[22] A tiny sampling of texts undergirded by this argument include Gillian Beer's "What's Not in *Middlemarch*," Eugene Goodheart's "The Licensed Trespasser: The Omniscient Narrator in *Middlemarch*," Brian Swans' "*Middlemarch* and Myth," Ralph W. Rader's "The Comparative Anatomy of three Baggy Monsters: *Bleak House*, *Vanity Fair*, *Middlemarch*," and J. Hillis Miller's "A Conclusion in Which Almost Nothing is Concluded."

Getting tangled in these two textual moments is easy for readers for both logistical and interpretive reasons. The opening prelude is the first moment in which readers can get a conceptual foothold in what is going to be, they already know, a long reading experience.[23] As a result, and compounded by a philosophical prelude that wonders about knowing the history of man, Eliot seems to be encouraging readers to latch onto a thematic lens with which they can read the entirety of the book. Similarly, by the novel's end, readers have spent numerous hours and, particularly if read in serialized form, many months reading a story in which nothing of traditional heroic or epic-level importance happens.[24] Eliot's finale provides a pacifying conclusion that gives readers the ability to easily explain the nothingness of the text they just completed as being representative of a world built primarily by forgotten nothingness.

My reason for questioning the relationship of these two narrative moments in the book, primarily embodied by the novel's Prelude and Finale, is that they seem incongruous with the rest of the text. Their styles, tones, subjects and structures do not belong. And yet, despite their obvious

---

[23] Although *Middlemarch* first appeared in serialized form, meaning the text itself would not have been as physically imposing as it currently is, the process of serialization meant that readers could expect an extended reading process. In this context, the book's opening prelude served as the foundation on which readers would have to store and access the text's information over the course of months.

[24] Bedient describes this sense of lacking epic-ness as a frustration of Eliot's, explaining that, "What George Eliot resents, what makes her strike out against phantoms, is Dorothea's lack of fame. Many who knew Dorothea, she remarks, 'thought it a pity that so substantive and rare a creature should have been absorbed into the life of another, and be only known in a certain circle as wife and mother.' To be only known in a certain circle! As wife and mother! The pangs, the emptiness, the comparative vulgarity of an unglorious life – it is this George Eliot regrets on Dorothea's behalf. Her pity flows unchecked and is almost palpable; for in Dorothea she sees her own worst fears enacted. Never will Dorothea's acts be classed as 'historic.' She will rest in an 'unvisited' tomb: so the closing words of the novel remind us, as if pointing out some sad and impressive truth" (79).

incongruity, the book's Prelude and Finale have profoundly guided *Middlemarch* interpretation and criticism.

Ralph W. Rader, for example, in a rebuttal of Henry James's assessment of *Middlemarch* as "a mere chain of episodes, broken into accidental lengths and unconscious of the influence of a plan," bases his entire argument around the novel's final sentence. He writes:

> But James made little effort to see what in fact Eliot's overall formal intention was or to gauge properly the artistic problems it presented. [...] But her purpose was clearly more definite than this, as is evident from the last sentence of her book. [...] Her formal intention, I believe, was to construct a plotted story the full significance of which this sentence would serve to render explicitly and complete. Her intention was not therefore to paint a static "social picture" but, as many have seen, to represent dynamic social process.[25]

Without Eliot's final sentence, the one Rader describes as the driving force behind the rest of the novel, is Rader able to make his argument? Does what "many have seen" in *Middlemarch* – a representation of a dynamic social process – remain the novel's primary theme?

More recent *Middlemarch* criticism continues to emphasize the importance of the novel's beginning and end, using it as a foundation on

---

[25] Rader, Ralph W. "The Comparative Anatomy of Three Baggy Monsters: *Bleak House*, *Vanity Fair*, *Middlemarch*." *The Journal of Narrative Technique*, Vol. 19, No. 1 (Winter, 1989). 62.

which to bolster the importance of us, *Middlemarch*'s readers. J. Hillis Miller, in a chapter emphasizing the results of the novel's conclusion, writes that:

> The ultimate paradox or contradiction in the finale is the double meaning it gives to the word and concept of "history." On the one hand, Dorothea's life is not historic. It could not find a place in any old-fashioned "monumental" history, with its tales of world leaders, kings, emperors, popes, battles, and wars. Herodotus can write about Cyrus but not about ordinary women of his time. On the other hand, real history is made by innumerable almost invisible acts such as Dorothea's "Oh, I cannot bear it. My heart will break."[26]

Miller's implication is that if "real history is made by innumerable, nearly invisible acts," then the largely insignificant acts performed during the lives of *Middlemarch*'s readers – our insignificant acts – are the makings of "real history."

Gillian Beer also makes *Middlemarch* a story about the importance of us, the readers, when, in another chapter of the book in which Miller's essay can be found, she explains how:

> The unhistoric and the unwritten, the unknown people who have read the book before us, temper the hubris of epic scope. *Middlemarch* makes its claim to inclusiveness by demurring at any all-embracing explanation, by offering us at last the sense of

[26] Miller, J. Hillis. "A Conclusion in Which Almost Nothing is Concluded." What's Not in *Middlemarch*." Middlemarch *in the 21st Century*. ed. Karen Chase. Oxford: Oxford University Press, 2006. 153.

things left out: an elegy for all those unknown others by whom at any time the single reader is surrounded.[27]

Beer's implication is that we, the readers, also comprise the "unknown others" who, ourselves, have unvisited tombs.

Even Karen Chase, the editor of the collection containing both Miller's chapter and Beer's chapter, supports this emphasis on reading *us* into *Middlemarch*. In her introduction to the essays of Miller and Beer and the other scholars contributing to the collection, Chase concluding that because we continue making real history despite our unvisited tombs, "even the most dour reader conveys the novel's hope simply by reading *Middlemarch* in our own time."[28]

I do not deny the validity of any of these kinds of *Middlemarch* interpretations. They seem both reasonable and readily available. What I do wonder is whether these interpretations would be made, or would be so convincing, without the Prelude and Finale.

I introduce my concerns about basing *Middlemarch* interpretations so heavily on the beginning and ending of the book because Eliot herself seems to encourage questions about the way she begins her Finale, writing:

> Every limit is a beginning as well as an ending. Who can quit young lives after being long in company with them, and not desire to know what befell them in their after-years? For the

[27] Beer, Gillian. "What's Not in *Middlemarch*." Middlemarch *in the 21st Century*. ed. Karen Chase. Oxford: Oxford University Press, 2006. 34.

[28] Chase, Karen, ed. Middlemarch *in the 21st Century*. Oxford: Oxford University Press, 2006. 14.

> fragment of a life, however typical, is not the sample of an even web: promises may not be kept, and an ardent outset may be followed by declension; latent powers may find their long-waited opportunity; a past error may urge a grand retrieval. (1183-1185)

With these words, Eliot gives readers an explicit warning that all limits, which, by necessity, must include the beginning and ending limits of *Middlemarch*, are artificial human constructs. They are "not the sample of an even web." Perhaps placing so much interpretive emphasis on beginnings and endings ignores their unnaturalness.

A brief look beyond *Middlemarch* suggests that the artificiality of limits is an important concern of Eliot's, that she struggled creating them, and that readers might want to be cautious when emphasizing them as the moral axis on which the novel rotates. Little critical attention seems to have been paid to this issue, particularly recently, but in a 1965 article titled "George Eliot's Conception of 'Form,'" Darrel Mansell, Jr. makes some observations that, judging by the common scholarly emphasis on *Middlemarch's* beginning and ending, scholars seem to largely disregard.[29] Mansell explains that:

> Throughout her career [Eliot] discourages the reader from giving as much significance to the beginning and ending as to the intrinsic relations of which these are only the outer limits.

[29] D.A. Miller, in an essay considering perspective in *Middlemarch*, makes reference to the same Mansell article and notes Eliot's struggle with conclusions, but the recognition does not seem to create any skepticism on Miller's part about how to value the book's conclusion. ("D.A. Miller on perspective and Dorothea at the Window." fr. *George Eliot: Comprehensive Research and Study Guide, Bloom's Major Novelists*. ed. Harold Bloom. pg 105)

> The first words of the first chapter motto of *Daniel Deronda* tell the reader that the poet can do nothing "without the make-believe of a beginning." She makes the point in a letter that "endings are inevitably the least satisfactory part of any work in which there is any merit of development" (*Letters*, VI.231-242). She speaks in a review of the "artificial necessities of a denouement." And she maintains that "conclusions are the weak point of most authors, but some of the fault lies in the very nature of a conclusion, which is at best a negation" (*Letters*, II.324).[30]

Eliot's above-noted comment about conclusions included in one of her letters is particularly illuminating. It suggests she likely had trouble writing conclusions, and it articulates her dissatisfaction with their artificiality. Mansell, by studying the progression of Eliot's novels, and by considering many of her letters and essays, ultimately determines that "at the end of her career [Eliot] had pressed her conception of 'form' so far that the beginnings and endings of her novels had become a source of frustration."[31] One short essay in particular, Eliot's "Notes on Form in Art," written close to the beginning of her work on *Middlemarch* in 1868, offers a helpful understanding of Eliot's beliefs about artistic structure which explains her dissatisfaction and struggles with the artificiality of limits.

---

[30] Mansell, Jr., Darrel. "George Eliot's Conception of 'Form'." *Studies in English Literature, 1500-1900*, Vol. 5, No. 4, Nineteenth Century (Autumn, 1965). 661.
[31] Ibid. 652.

Eliot's "Notes on Form in Art" essay outlines not just her conception of form as it relates to art, but also how Eliot perceives the structure of the world. The essay reveals how the *Middlemarch* conceit of a "web" is more than just a belief of the book; the concept of the interconnected web appears to be, in Eliot's view, the nature of the universe. Eliot argues that:

> Plain people, though indisposed to metaphysical subtleties, can yet understand that Form, as an element of human experience, must begin with the perception of separateness, derived principally from touch of which the other senses are modifications; and that things must be recognized as separate wholes before they can be recognized as wholes composed of parts, or before these wholes again can be regarded as relatively parts of a larger whole.[32]

People, according to Eliot, need to perceive boundaries in order to comprehend the world in which they live, but those boundaries are only applied for convenience and are not actual boundaries. The result of creating these artificial boundaries is what Eliot deems a "necessary antithesis," which is a "sense of wholeness or unbroken connexion in space and time [*sic*]."[33]

Because Eliot perceives limits as necessities of human experience needed to help people process the wholeness of the universe, form becomes an artificial and constricting construct. "Boundary or outline and visual appearance are modes of Form which in music and poetry can only have a

---

[32] Eliot, George. "Notes on Form in Art." fr. *George Eliot: Selected Essays, Poems, and Other Writings*. New York: Penguin Classics, 1991. 231-232.
[33] Ibid. 232.

metaphorical presence,"[34] Eliot explains. "What is a structure but a set of relations selected and combined in accordance with the sequence of mental states in the constructor, or with the preconception of a whole which he had inwardly evolved?"[35]

Eliot's "Notes on Form in Art" underscore the author's personal struggle with beginning and ending her texts because she believes such constraints of form are unnatural. Her belief in this unnaturalness, and her struggle with the resultant boundaries of texts, seems to challenge arguments about *Middlemarch* that rely on the book's artificial – by Eliot's definition – beginning and end. Eliot's beliefs suggest she would have struggled writing those parts,[36] and her beliefs suggest those parts would not have contained the book's most reliable passages for use as textual ciphers.

Instead of aggrandizing two brief textual moments that are entirely disparate from anything else in the book, instead of disregarding Eliot's personal beliefs about the artificiality of limits, and instead of turning the limits of *Middlemarch* into the novel's moral and philosophical cipher, I wonder what happens if we ignore them – at least for now – even if doing so

---

[34] Although Eliot specifically uses the word poetry, she is likely considering any forms of artistic language, including novels. She explains this interpretation of poetry a few sentences later when she writes: "My concern is here chiefly with poetry which I take in its wider sense as including all literary production of which it is the prerogative and not the reproach that the choice and sequence of images and ideas – that is, of relations and groups of relations – are more or less not only determined by emotion but intended to express it." (233)

[35] Ibid. 233.

[36] Much has been written about Eliot's production process for *Middlemarch*, and specifically how she began with her "Study of Provincial Life" before stopping work on the project for over a year (in part do to the death of the George Henry Lewes' – her companion's – son). Eliot would ultimately reengage the project by starting work on a separate book, "Miss Brooke," that would eventually become the beginning of *Middlemarch*. A more detailed explanation of how Eliot composed *Middlemarch* can be found in Jerome Beaty's Middlemarch*: From Notebook to Novel*, and John Clark Pratt and Victor A. Neufeldt also provide helpful transcriptions of Eliot's *Middlemarch* Notebooks.

carries the risk of turning *Middlemarch* into a text which is, as Henry James's previously-noted critique famously and controversially asserts, "a mere chain of episodes, broken into accidental lengths and unconscious of the influence of a plan."[37]

Interpreting *Middlemarch* while ignoring the Prelude and Finale is not as daunting as it might, at first, appear. In addition to all the evidence regarding her dislike of limits, Eliot provides plenty of resources, both within the novel and in her writings outside of it, to guide alternate readings. Returning, first, to her "Notes on Form in Art," Eliot explains how something that might seem like a chain of unrelated, disparate episodes actually combines to form a more complex whole. She writes that:

> As knowledge continues to grow by its alternating processes of distinction and combination, seeing smaller and smaller unlikenesses and grouping or associating these under a common likeness, it arrives at the conception of wholes composed of parts more and more multiplied and highly differenced, yet more and more absolutely bound together by various conditions of common likeness or mutual dependence.[38]

For Eliot, the seemingly individual parts should not be read as individual at all because they are intimately connected to everything surrounding them. Those connections result in a complex and more realistic formation of a whole. The

[37] James, Henry. fr. Review in *Galaxy*. ed. William Conant Church. Vol. 15 (Jan. 1873 - Jun. 1873). 425.
[38] Eliot, "Notes on Form in Art," 232.

connections are what create Eliot's metaphor of a web in which the individual strands might seem, by themselves, frail, but, when woven together, they make a unified, versatile, sturdy piece of fabric.

Recognizing that the distinct episodes of *Middlemarch* are being interwoven into a unified mesh is an understanding of the novel that, for some readers, might not require George Eliot's explanation. All the stories are within the (artificial) form of a single novel, so presumably they have a relationship. However, what becomes difficult, particularly without interpreting the novel through the lens of the Prelude and Finale, is determining how the seemingly disparate stories relate to one-another. But we can better understand their relationships thanks to an excerpt from "Notes on Form in Art" where Eliot offers a helpful clue about fiction, telling readers:

> Sometimes the wider signification of poetry is taken to be fiction or invention as opposed to ascertained external fact or discovery. But what is fiction other than an arrangement of events or feigned correspondences according to predominant feeling? We find what destiny pleases; we make what pleases us – or what we think will please others.

Eliot is arguing that despite being perceived as invention, and despite its contrast to the things in this world considered to be "real," fiction influences readers by operating within communal senses of feeling. Those feelings, in this case, are not limited to feelings in an emotional sense, but also include feelings of a predominant sense of reality. By tapping into this shared sense of reality, the novel is a product of that reality and is as successful a conveyor –

or as unsuccessful a conveyor – of what is real and factual as anything and everything else. Fiction, according to Eliot's assertions, appears to be a mechanism for storing and disseminating shared feelings. In this regard, fiction could be operating as a technology-enabled information space, usable by its readers to connect with shared cultural knowledge.

Recognizing "fiction" and "poetry" (both of which, in Eliot's essay, are references to any artistic literary text) as connective technologies for accessing information spaces is more than just an approach to reading and interpreting *Middlemarch*; it is an approach to literary texts that permeates the novel itself. Throughout the novel, Eliot uses fiction and literary texts in precisely this way. In order to help readers better process her own text, Eliot leverages the ability of other literary objects to connect readers with remotely stored cultural information.

While my specific focus is Eliot connecting with literary texts, I should first note that Eliot connects *Middlemarch* readers to far more than just external literary knowledge. In addition to Middlemarch's literary connections, many critics have also discussed the novel's relationships with science, myth, politics, philosophy, and other types of textual and cultural knowledge. To even attempt a reasonably inclusive discussion of either the references Eliot makes or the critical discussions of them would likely result

in a document as lengthy as *Middlemarch* itself.[39] Even the task of thoroughly considering only Eliot's literary allusions could be the work of hundreds of pages. By my admittedly unscientific count, I noted roughly 200 specific and general literary references in *Middlemarch*, and I suspect my count to have missed many of Eliot's more subtle allusions.[40] Instead of attempting any sort of unachievable comprehensiveness of discussion regarding these textual connections, I will avoid what Eliot might refer to as "that tempting range of relevancies called the universe" and, for many of the same reasons Eliot limits her own scope, I will concentrate all the light I can command on one connective literary link within *Middlemarch*'s particular web.

The illustrative example of a connective literary link in *Middlemarch* on which I will focus appears in the narrative preface leading to Eliot's often-discussed "particular web" comment. Eliot has just concluded Chapter XIV, in which she narrates the visit of Fred to his uncle, Featherstone, where Fred is hoping for money he can use to repay a large debt. In Chapter XV's transition back to the story of Lydgate, Eliot's first paragraph – the paragraph that precedes her famous quote – connects her novel with the noted 18th century English dramatist and novelist Henry Fielding. In it, the narrator juxtaposes Fielding's narrative style "where he seems to bring his armchair to the proscenium and chat with us in all the lusty ease of his fine English" (1948) with the narrator's own inability to "linger after [Fielding's] example" (1951)

[39] Eliot's surviving *Quarry* (transcribed by Anna Theresa Kitchel in *A Quarry For* Middlemarch) provides a useful starting point for exploring the kinds of research she did prior to and while writing *Middlemarch*, and, as a result, the kind of disciplines linked within the text.
[40] Again I will point to the above-noted *Quarry* in order to demonstrate just how broad and inclusive Eliot's knowledge base is, making the possibility of recognizing every allusion and reference nearly impossible.

due to constraints of time. Despite the seeming fact that Eliot's narrator is contrasting his style with Fielding's, Eliot might actually be using the Fielding reference as a resource with which readers can better understand the evolution of *Middlemarch*'s structure.

This possible link to Fielding can be better understood by using one of Gillian Beer's explanations for how *Middlemarch* uses a web-like structure. Beer explains that:

> Multiple analogy creates a web of interconnections in *Middlemarch*. George Eliot's intellectual characters are preoccupied with sources: 'the primitive tissues', 'the key to all mythologies', but the text is organised in terms of variability. *Middlemarch* creates an experimental situation by its use of structural analogy and by its 'provisional framing', which draws the focus ever more sharply, shifting and refocusing where necessary, testing situations through diverse consciousnesses, repudiating the subjectivity of the single point of view. [*sic*][41]

Instead of perceiving the Fielding reference as a contrast to show how Eliot's and Fielding's styles differ, the use of analogy Beer describes, which pervades the entirety of the novel, suggests that even though the Fielding-Eliot juxtaposition is presented as contrastive, it is far more comparative. The analogy, in this case, demonstrates how, despite Eliot's narrator claiming to be only focused on a small "study of provincial life," he is actually, like

[41] Beer, *Darwin's Plots*, 154.

Fielding with all his time for lingering on history, discussing the larger universe.

This recognition that *Middlemarch* is, indeed, connecting people to information about the larger universe comes at a particularly important moment in the evolution of human knowledge. As much of Beer's work explains, Eliot is publishing at a time when Charles Darwin's theories seem to be requesting radical shifts in mankind's understanding of his place in the universe. This struggle is summarized well when Beer writes:

> In a world which no longer consisted of fixed species a struggle had begun to manifest between external form and potential meaning: 'she was undergoing a metamorphosis', she writes of Dorothea, 'in which memory would not adjust itself to the stirring of new organs'. Relationships become the organising principle of all life and in *Middlemarch* this is emphasised in repudiation of any search for origins or even of succession. *Middlemarch* is a single work preoccupied with the 'web of affinities', setting out relations in a particular space and time. This allows a sense that everything is knowable and even that it may finally become known. [*sic*][42]

Middlemarch's cultivation of the sense that everything is knowable and that it might soon become known is embodied specifically in Eliot's ability to connect with Fielding. By connecting with Fielding, Eliot, through the work of analogy that persists throughout her book, is showing that the knowledge

[42] Ibid. 154.

being cultivated by that "great historian, as he insisted on calling himself" (1945) is not being lost. By linking to Fielding and creating the analogy between her work and Fielding's, Eliot demonstrates both an evolution of texts and how texts, despite such evolutions, continue to be, for their readers, what they always have been: information spaces preserving and building upon pre-existing knowledge about "that tempting range of relevancies called the universe."

The ability of literary texts to preserve existing knowledge and build new knowledge by linking with other, separately created and maintained literary texts should, in itself, resemble an instructive analogy. After all, Tim Berners-Lee's World Wide Web was built to accomplish the same goal (i.e. preserve knowledge in order to build new knowledge) through the same methodology (i.e. the linking together of separately created and maintained documents built using the same protocol). Although the technology of print media is not capable of providing immediate access to linked texts, the implementation of the links themselves – the interconnection of information spaces – is still a fundamental principle empowering the ways in which a technology like a novel connects, preserves, and transmits information. Martin Mueller offers a succinct way of explaining this operating structure for the technology of books when he writes, “A book sits in a network of transactions that involve a reader, his interlocutors, and a ‘collective library’ of things one knows or is supposed to know.”[43]

[43] Mueller, Martin. “Digital Shakespeare, or towards a literary informatics.” Shakespeare, Vol.4, No. 3, September 2008. 9-10.

Consider, briefly, some other notable examples of Eliot employing textual linking in order to expand the range of information about a particular object or concept. The most obvious of these examples are likely Eliot's epigraphs beginning each chapter, which are sometimes, though not always, quotes from other authors including Miguel de Cervantes (II), Milton (III), Virgil (IV), Shakespeare (XXVI), and Spenser (XXXVII). Michael Peled Ginsburg, in "*Middlemarch* and the Problem of Authorship," explains that:

> The use of epigraphs establishes a relation between the text and a past tradition. That relation can be one of illustration, in which the text repeats or exemplifies the insights of the epigraph, or it can be an ironic relationship, in which the text calls into question the truth of the epigraph. In both cases, however, the result of the act of preceding the text by an epigraph is that the text defines itself in relation to a past text, thus pronouncing itself secondary and derivative while positing the epigraph-text as prior and original.[44]

The derivative nature Ginsburg refers to is representative of how, through textual linking, Eliot is using the technology of the novel to preserve information. By linking to other authors, and by positioning her work as derivative of those authors, Eliot directly demonstrates the ability of prior texts to store information and her ability to both access and respond to that

[44] Ginsburg, Michael Peled. "Pseudonym, Epigraphs, and Narrative Voice: *Middlemarch* and the Problem of Authorship." *ELH*, Vol. 47, No. 3 (Autumn, 1980). 546.

information in generative ways capable of producing and transmitting new information.

While epigraphs are a prominent way in which Eliot uses the technology of previous textual traditions as contextualizing links, she also includes textual links throughout the narrative itself. Some of the more apparent examples of employing literary links within the narrative are the many direct and indirect references to John Milton. The image of Milton, the blind scholar-poet who, in his later years, conscripted his daughters to serve as his scribes, is often intertextually linked with the relationship between Dorothea and Casaubon. By linking to the story of Milton, Eliot does not need to construct an entire frame for Dorothea and Casaubon's relationship. Instead, she can build upon a pre-established textual and historical tradition in order to situate her own characters. As a result, when Eliot writes: "Dorothea had thought that she could have been patient with John Milton, but she had never imagined him behaving in this way" (4020-21), in the span of a mere 22 words Eliot infuses *Middlemarch* with generations of Miltonic textual and interpretive details.[45]

Not all instances of textual linking within *Middlemarch* are as overt as epigraphs or the specific naming of Milton. For example, readers have often

[45] The link between Milton and Eliot is far more rich than this simple example can explain. A more thorough discussion of the relationship Eliot has with Miltion is available in Anna K. Nardo's *George Eliot's Dialogue with John Milton*. In addition, the ability to introduce Miltonic overtones is not limited to Eliot or *Middlemarch*. By including Milton here, as well as this footnote's link to a text about the relationship between Milton and Eliot, we might even wonder whether this document is leveraging a similar kind of narrative linking technology.

found embedded in *Middlemarch* the influences of Shakespeare.[46] Joseph Allen Boone describes *Middlemarch* as "one of the last great marriage novels to conform to the Shakespearean dictum that 'journeys end with lovers meeting.'"[47] Marianne Novy, in an article titled "*Middlemarch* and George Eliot's Female (Re)Vision of Shakespeare," makes numerous links between *Middlemarch* and Shakespeare, including describing Mary and Fred's courtship as a Shakespearean comedy, paralleling Dorothea and Desdemona, and ultimately suggesting that Eliot could claim Shakespeare as a literary ancestor "without giving up [her] interest in women's experience."[48] And Otice C. Sircy employs Shakespeare's sonnets to understand Eliot's representations of love. Sircy writes:

> In their falsification of nature, the sonnets offer a resonant paradigm for the egotistical element in love which helps to shape our response first to Casaubon and then to Ladislaw in the first half of the novel. With Casaubon who momentarily wraps himself in the borrowed robes of a courtly lover, the falsification becomes the source of an increasingly restrictive imprisonment of the self. With Ladislaw who indulges an inclination "to love what is good and beautiful when [he sees] it" (ch. 39) by elevating Dorothea to that pedestal where "a creature

[46] Eliot employs five epigraphs she attributes to Shakespeare, but she never directly mentions him in the novel's narrative. Even a nominal reference to "the poet" by Mr. Trumbull during the auction of a portrait (8706) is not a reference to Shakespeare, but a reference to Thomas Gray.
[47] Boone, Joseph Allen. *Tradition Counter Tradition: Love and the Form of Fiction*. Chicago: University of Chicago Press, 1989. 141.
[48] Novy, Marianne. "*Middlemarch* and George Eliot's Female (Re)Vision of Shakespeare." *The Journal of English and Germanic Philology*, Vol. 90, No. 1 (Jan., 1991). 64, 70, and 78.

> worthy to be perfectly loved" is enshrined in the convention of the sonnets, the falsification is an entrapment as much of the eyes as of the emotions.[49]

Regardless of whether or not an argument like Sircy's has any validity (a discussion not relevant to my current analysis), the fact that a reader of *Middlemarch* leverages Shakespearean sonnets to better understand the novel is a testament to literary technology's ability to preserve, transmit, and connect people with information. Sircy is connecting with *Middlemarch* through Shakespeare and connecting with Shakespeare through *Middlemarch*.

Since I am claiming that literary technologies preserve, transmit, and connect people with information, a concern with this assertion might be to wonder what kind of information a text like *Middlemarch* contains. If not, what is the purpose of storing the information in the first place? But directly accessing a literary text's stored information seems to reveal one of literary technology's most problematic shortcomings as a shared information space. While literary technology admirably preserves and transmits information, accessing that information is often challenging; the information itself is, in a sense, encrypted. Though easy to overlook, the system of encryption used to encode literary technology has a name: ‘art.’ Art is the culturally accepted name given to technologies of communally accessible information spaces that

[49] Sircy, Otice C. "'The Fashion of Sentiment': Allusive Technique and the Sonnets of *Middlemarch*." *Studies in Philology*, Vol. 84, No. 2 (Spring, 1987). 243.

do not make their information readily available. In order to access the information, the user must decrypt it.

If applying the concept of encryption and decryption to the notion of artistic merit seems like too far a gap to reconcile, that concern might be a result of the relative youth of the terms. The words *encryption* and *decryption* are products of the 1950's and the birth of the digital age, but the concepts themselves are much older. Encryption is a process of converting information into an alternate form. As a result, accessing the original information becomes a process of decrypting the information in its new form back to its original. Since literary texts are information spaces, one of two things must be true: either a text like *Middlemarch* is simply a story about provincial life with no additional messages embedded or implied; or *Middlemarch* is using the story of Dorothea, Lydgate, Rosamond, Fred, and the rest to encode, store, and transmit information about things like the interconnected web of human experience, the ability of literary texts to transmit information, or even the importance of people "who lived faithfully a hidden life, and rest in unvisited tombs." If the latter is true – and textual critics seem to agree it is since none read *Middlemarch* without considering meaning beyond narrative plotlines – then literary texts use artistic representation as a device for encrypting information.

Recognizing *Middlemarch* as cultural information encrypted by an artistic cipher allows readers to analyze the text on its own terms. Specifically, in order to decrypt an encrypted object, its cryptographer must rely on a key. Similarly, in *Middlemarch*, using a key to unlock important information is a

pivotal theme. At its most literal point, Featherstone dies while grasping for the key he needs to unlock a chest containing his will. At its most metaphoric, Casaubon dies while searching for his imagined Key to all Mythologies.[50] But other figurative and literal keys are needed to unlock information as well. They include the key Dorothea needs to open the jewelry box containing her mother's old jewels,[51] and they include the key to Bulstrode's wine-cooler, where the alcohol in it ultimately leads to Raffles' death and the public unlocking of the information about Bulstrode's past.

Just as keys are important for unlocking information about and for characters within *Middlemarch*, readers also need keys to unlock the cultural information preserved and transmitted by *Middlemarch*. Without those keys, *Middlemarch* is little more than a well-formed story about a small, English town. But with those keys, *Middlemarch* becomes a technology of information storage and dissemination.

Just like Dorothea needs a key upon her return to her Lowick home in order to examine her late husband's papers and unlock the story of her own web (6958), readers of *Middlemarch* need a key (or keys) to unlock the book's textual web. George Eliot seems aware of this need, and one way of providing keys could be by creating parallel characters. On one side of the parallel are all the characters driven by an egoism that convinces them they deserve to have control over their legacies. For guidance on how to leave a legacy, they ignore

[50] The futility of Cauaubon's search might even be a way of expressing another shortcoming of literary technology's ability to store and transmit information. The imperfections of the technology make complete decoding impossible.

[51] Nina Auerbach describes the information retrieved by unlocking the jewels when the writes that: "Family jewels are a clasp from the past; they are far from Heaven, but through them, we touch the flesh and spirit of lost ancestors" ("Dorothea's Lost Dog," 88).

the needs of the present as they explore the past to see what others who have left their marks on history have done. The history-focused characters most obviously include characters like Casaubon and Dorothea and Lydgate, but they also include, in more subtle ways, characters like Bulstrode, with his need to build the hospital, and Featherstone, with his emphasis on manipulating his familial legacy. Those characters all fail to leave the legacies they intend because of the unavoidable flaw in such a pursuit: legacies are not controlled by the people who die; they are controlled by the people who live.

On the other side of the character parallel that helps unlock Eliot's text are the characters who realize the past is not a true representation of the past, but instead, what is perceived as the past is actually a representation of the present. They are the characters who, as Hilary Mackie suggests in "The Key to Epic Life? Classical Study in George Eliot's *Middlemarch*," are able to "recognize [that] the images and ideas from antiquity are only usable insofar as they are altered and adapted to suit new present-day contexts," and they include characters like Mr. Farebrother and Mary Garth, who question and adapt the past.[52] These character parallels suggest that one key for successfully unlocking information preserved within artistic technologies of information storage and dissemination, be they textual, oral, mythical, religious, or digital, is to be able to contextualize those technologies during the moments in which they are being accessed.

[52] Mackie, Hilary. "The Key to Epic Life?: Classical Study in George Eliot's *Middlemarch*." *Classical World* 103.1 (2009). 66.

Such is the case for a book like *Middlemarch*, where its usefulness today is not the same as its usefulness in the 1870s when it was first published. And its usefulness 100 years from now will presumably be substantially different than how I use it today as a parallel with digital information spaces.

Unlocking the information within texts by means of the historical moments in which they are accessed enables texts to perform the work Tim Berners-Lee, when he proposed the need for the World Wide Web, argued printed media could not do. He argued that printed texts were unable to be updated quickly enough to keep pace with the ever-evolving state of information. Although he was technically correct when he argued that printed texts cannot be physically updated quickly (through re-printing), he appears to have been wrong in assuming they cannot easily adapt to contain current information. Because print media was, prior to the World Wide Web, the best tool available for creating shared information spaces, its most successful users seem to have been to ones who learned to give printed texts dynamic updateability. They could do so, as Eliot demonstrates, by creating artistic texts capable of transferring culturally-preserved information through a historically-contextualizing process of decryption.

The reason for artistry in textual creation – perhaps the reason why people like George Eliot wrote novels instead of treatises in order to preserve and transmit shared cultural information – is that the artistry of the medium enables dynamic adaptability. As a result, the problem Tim Berners-Lee

describes of print media being an insufficient tool for preserving and transmitting the vast amounts of information being constantly produced and accessed is a problem that can be circumvented by encrypting information within artistic narratives like – but not limited to – novels. Through the work of novels (and presumably through other artistic media), an author allows readers to extract from a text not just shared information, but the information contextualized and made relevant to the historical moment in which it is extracted. Artistic production, in this view, becomes a process for making information dynamic.

By adopting this view of understanding artistry as a process of making information dynamic, the successes and failures of artistic productions could be judged based on their ability to transmit information relative and applicable to the historical moments in which they are accessed. In the case of *Middlemarch*, for example, if the novel is a successful artistic text, then this judging rubric would define it as successful because the information it contains is as relevant and beneficial to the work of its readers a century ago as it is to readers today. In addition, for it to remain a successful artistic text, its shared information space must be useful to the work of its readers in the coming centuries.

With this approach for considering how technologies like novels preserve information, I propose reconsidering the Prelude and Finale of *Middlemarch* not as beginnings and endings, but as two pieces of the text helping to connect *Middlemarch* to the historical moment in which it is being accessed. Eliot seems to present this kind of link by creating a Prelude and

Finale that concern themselves not with the story of provincial life in early 19th century England that occupies the majority of the text, but instead the Prelude and Finale concern themselves with history and the passage of time. By bookending her story with concerns about history, Eliot is allowing readers to extract the events of *Middlemarch* from their 19th century motif and make them indicative of events in any reader's lifetime. Because of this fusing of the novel's time and the reader's time, the reference to people "who lived faithfully a hidden life, and rest in unvisited tombs" does not need to only embed the novel with the sense of ultimate despair with which many people seem to read it.[53] The unvisited tombs in the novel's closing words can also be understood as unvisited because, in the context of the metaphor, they need not exist. Instead, the ability to preserve, transmit, and access the knowledge built by past generations through artistic storage technologies could mean that nothing ever dies. Things that might otherwise seem lost – like information, events, and people – are maintained and accessible in shared, dynamic information spaces like novels.

Then again, maybe the closing sentence of *Middlemarch* should be read as others understand it. Perhaps Eliot means to remind readers of the unvisited and insignificant tombs of so many other people in history, and how unvisited tombs are the likely destinations for our own lives. If that is the case, then my point is only further confirmed. The very dynamism,

[53] Nina Aurbach best exemplifies this tendency when she claims that, "Death, not any one narrative, is the star of all these books, blanketing all stories not as life's end but as its primary condition" (104).

updateability, and adaptability inherent in the technology of the novel are the artistic encryptions that allow for alternate readings and interpretations.

When considering new technologies of information preservation, the novel's ability to be understood in multiple ways should not be overlooked. And yet, that seems to be happening. Or, if not specifically overlooked, digital technologies are, at the very least, diminishing people's appreciation for the novel's ability to contain dynamic information. By overcoming some of printings limitations of information preservation and dissemination, the World Wide Web could be perceived as eliminating the need for artistic textual production. After all, why rely on artistically encrypted texts that require non-standardized temporal-interpretation in order to access dynamic, shared information when the Web allows users to connect with and update shared information spaces immediately? But that question is not the question I am most curious about. History has long since established natural procedures for the introduction and incorporation of new information technologies, and establishing a relationship between the medium of the novel and the medium of the Web is one I will leave history to decide. Instead, I recognize how, despite the improvements the World Wide Web makes to information preservation and information dissemination, it is still, just like print media, not a perfect technology of storage and transmission. Those imperfections will need to be overcome, and I suspect, like with print media, they will be overcome through artistic innovation.

Chapter 5: Emily Dickinson, Software Engineer

"Code is poetry" is the slogan of WordPress. Originally launched as open source blogging software by Matt Mullenweg and Mike Little on May 27, 2003, WordPress is, as of this writing, one of the world's most used web development platforms and content management systems, and it is currently deployed on millions of websites. Its purpose is to enable the easy uploading of content to a website through an administrative dashboard. The result is that users can publish content on the Web without needing to alter a website's codebase or know how it operates. If code is indeed poetry, the poetic-ness of the WordPress codebase might be described as the code's ability to eliminate a user's need to read, understand, and manipulate code.

But "being poetic" does not unequivocally equate to "being poetry." Although, as both a software developer and a longtime student of poetry, my inclination toward poetic-ness makes me eager to adopt the "code is poetry" mantra as a sort of rallying cry around which all my academic and digital work can safely muster, as I consider the WordPress slogan, I find myself having to pause and carefully consider the two "poetic" media types. I can describe plenty of things aside from code and poetry as poetic – the flight of an eagle, Dvorak symphonies, sunsets – but just because something has poetic qualities, is it necessarily poetry? I ask this question because, while I recognize that code can have poetic qualities, and, as someone who codes, I am eager to elevate the work of a developer to the work of poet, I find myself resisting the WordPress assertion that code is poetry.

To explain my resistance, I want to offer an explanation of poetry offered by Muriel Rukeyser. Rukeyser writes that:

> If there were no poetry on any day in the world, poetry would be invented that day. For there would be an intolerable hunger.
> And from that need, from the relationships within ourselves and among ourselves as we went on living, and from every other expression of man's nature, poetry would be – I cannot here say invented or discovered – poetry would be derived.[1]

Rukeyser is arguing that poetry is so imperative to the very existence of humanity that even if it did not exist it would be made to exist. While a person might choose to agree or disagree with Rukeyser's assertion, for my purposes, the existence of the assertion itself fosters my skepticism of the claim "code is poetry." In comparison, similar assertions concerning the inherent necessity of code within the world are not common. Even WordPress, with its prominence among the web development community and its "code is poetry" slogan, does nothing to make a case implying the world's "intolerable hunger" for code. The "about" section of the WordPress.org website, for example, says nothing of code's societal importance, but instead focuses on the software's development history and practical utility.[2]

But the WordPress slogan does suggest a relationship between code and poetry, and the primary use of WordPress as a blogging and web development platform expands on why that relationship may be worth

[1] Rukeyser, Muriel. *The Life of Poetry*. Ashfield, MA: Paris Press, 1996. 159-160.
[2] "About Wordpress." WordPress.org. Web. 19 Mar. 2014. http://wordpress.org/about/.

exploring. WordPress is a technology of information storage and dissemination built with code. If, as this project has argued throughout, the technology of literature also acts as a kind of cultural information technology – a technology constructed with language – then WordPress and poetry should have similar information management principles. Because of this connection, analyzing some of these similarities in conjunction with understanding how the WordPress codebase facilitates the storage and dissemination of information could suggest ways in which the same tasks are accomplished with poetry.

In order to analyze this connection in more detail, I propose ignoring – at least for now – any question of whether or not code is poetry. In its place, I propose considering the question in reverse: is poetry code? The advantages of the reversed form of the question is that, in comparison to poetry, the expectations of what code is, should be, and should do are relatively stable. As used in the context of software development, "code" is the symbolic representation of programming languages created for the purpose of instructing machines. With this definition of code readily available, we can then ask: Is poetry a symbolic representation of programming languages created for the purpose of instructing machines? To explore this possibility, I will engage the writings of the nineteenth century American poet Emily Dickinson as though she were a software engineer writing code.[3]

[3] Although the analysis that follows is primarily an interpretive engagement of Emily Dickinson's poetry, I should note that a substantial argument is available regarding the ways in which Dickinson's poetic production process also appears to have been programmatic in nature. While this chapter will primarily focus on an interpretive and syntactic analysis, Sharon Cameron's work on Dickinson's poetic variants and the generative possibilities that may have surrounded

***

In order to lay the foundation for analyzing Dickinson's poetry as code, I will return, again, to WordPress. Despite being a Content Management System (CMS) designed to minimize the need of knowing code for the average user, WordPress, itself, is open source software written in the PHP programming language. While other software could illustrate the programmatic concepts I will be discussing, WordPress serves as a particularly good first entry point for the coming analysis of Dickinson's poetry because of its central piece of programming logic known to WordPress developers as "The Loop."

The WordPress Loop facilitates the platform's primary content presentation structure. Although WordPress has become a versatile web development platform, it began as a simple blog management software. As such, the core functionality of WordPress is to help blog operators easily publish new content on their blogs in a form commonly referred to as a "post." When a blogger publishes a new post, that post becomes the first post on the blog, with all previous posts being displayed after, usually in reverse

---

their creation would be a good starting point from which to build a production-themed argument. Cameron explains, for example, that, "The variants can be understood by the supposition that what is being developed by Dickinson in her maturity is a poetry that depends on variants which extend a single utterance, conceived as a unitary text, outward into the margins and downward through the fascicle sheet" (*Choosing Not Choosing*, 13-14). This kind of explanation for Dickinson's variants suggests other ways she is working against the natural limitations of a static-text medium to help create a programmatically dynamic text.

Another opportunity for analyzing the programmatic nature of Dickinson's development process comes from Susan Howe's *My Emily Dickinson*. In considering how Dickinson may have engaged with the texts of others, Howe suggests that Dickinson would have constantly asked herself the question: "How do I, choosing messages from the code of others in order to participate in the universal theme of Language, pull SHE from all the myriad symbols and sightings of HE?" (17-18). This kind of questioning implies that Dickinson viewed other texts as information storage and transmission technologies she could integrate into her own poetry – similar to programmatic APIs or libraries – during her process of textual production.

chronological order. The WordPress Loop is the programmatic structure responsible for displaying those posts. It operates by querying the website's database (where the post content is stored after being inputted by the blogger), and then it embeds the relevant information, such as post title, post content, post author, and post date, within a templated page layout. The code for the post layout is written inside of the coding logic comprising the WordPress Loop, and the result is that the software cycles through (i.e. "loops" through) each post in the database, and, based on the conditions of the blog, such as how many posts to show per page or what category of posts to show, displays the posts one after another using a repeated layout template. The following diagram (Figure 7) provides a visual representation of this looping structure:

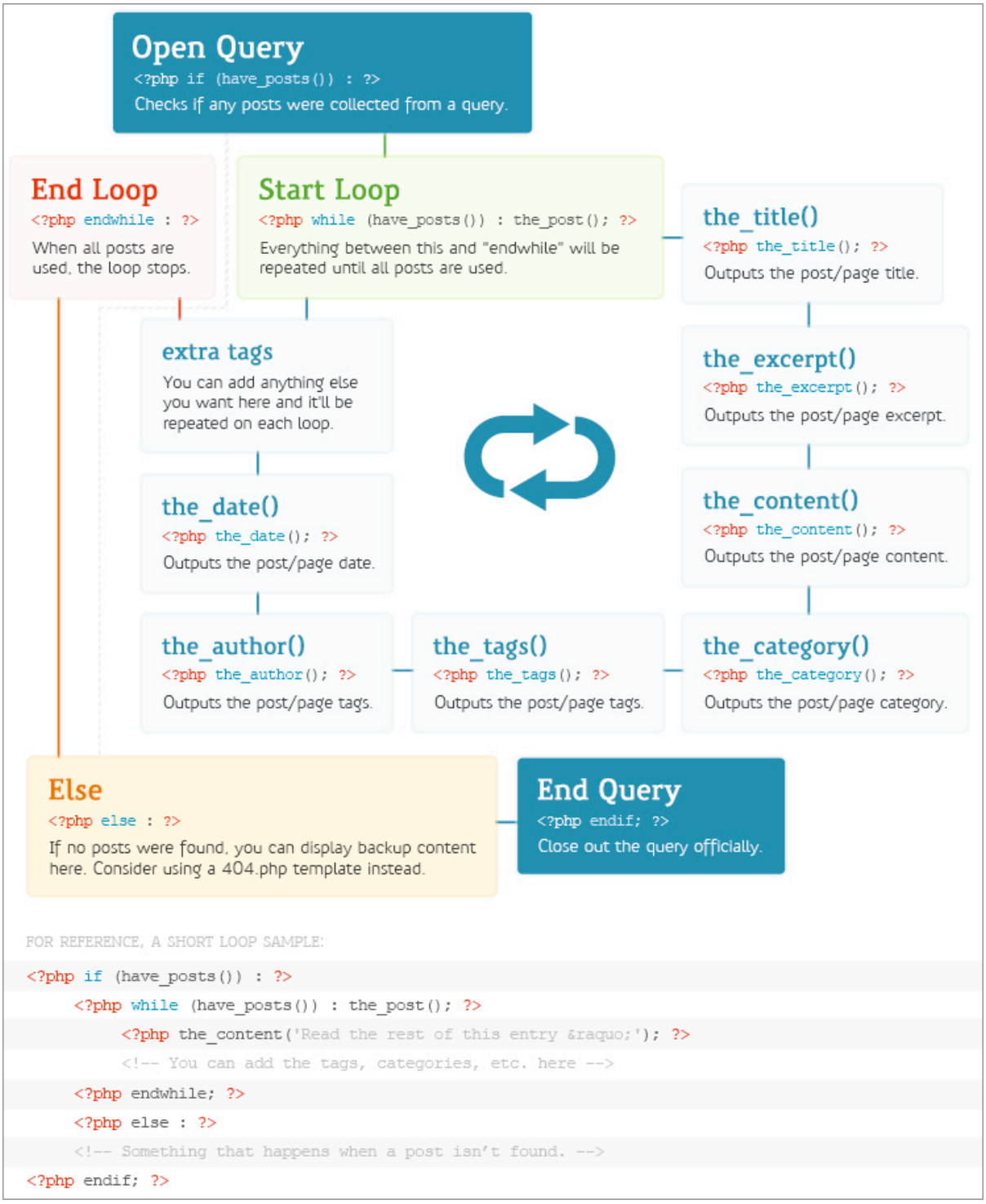

Figure 7: Diagram of the WordPress Loop created by Brandon Jones for tutsplus.com. It is most easily read in a clockwise pattern, starting with the "Open Query" conditional.

Most programming languages include the concept of a loop, and its basic logic is to instruct the processing machine to repeat a task until a certain condition is met. WordPress's display loop, when being used to display multiple blog posts, provides a good example of how the logic of a loop is represented in terms of repetitive content output. It also provides syntax that can serve as a first entry point into Dickinson's poetry because of similarities in diction between WordPress's underlying codebase and the Dickinson's poetic syntax. Those similarities stem from the kind of loop employed in the WordPress Loop. In programming terms, the WordPress Loop is known as a "while" loop. Using the keyword "while," the loop instructs the computer to cycle through a list of blog posts as queried from the database, and, *while* it is true that another blog post is returned by the query, the loop will execute all the code within the loop another time (i.e. it will display the templated information for another post) and then check again to see if yet another blog post needs to be displayed.

In computer programming, the concept of the "while" loop is a simple way to instruct a machine to repeat an action multiple times so long as certain conditions are met. Its prose translation is: "While X is true, do Y." But the concept is not unique to computer programming languages – a fact Emily Dickinson demonstrates in "While it is alive," where she writes:

While it is alive
Until Death touches it
While it and I lap
one Air

Dwell in one Blood

Under one Sacrement

Show me division can

split or pare –

Love is like Life – merely

longer

Love is like Death, during

the Grave

Love is the Fellow of

the Resurrection

Scooping up the Dust

And chanting "Live"!

(FP287, Appendix 1)[4]

[4] For Dickinson poems represented in this chapter, I am reproducing their words, as best as possible, using the basic lineations and structures within Dickinson's manuscripts. But I should also note the challenges of reproducing Dickinson's texts in print because, as Ellen Lousie Hart and Martha Nell Smith explain in the introduction to *Open Me Carefully*, "Dickinson used the page itself, and the placement of words in relation to embossments, attachments, and margins to convey meaning, and in ways that typography cannot sufficiently transmit" (XXIII). Because of these limitations of my text-based presentation technology, Appendix 1 includes URLs for more detailed viewing of the manuscripts online in the *Emily Dickinson Archive*. I am including these links in order to introduce some of the opportunities for analysis not available with transcriptions. Chief among these opportunities is to explore how Dickinson's use of the page evokes questions about the ways in which she engaged the limitations inherent in her choice of information technologies (i.e. handwriting, paper, pen/pencil). Aside from the Cameron text referenced in an earlier footnote, a beginning and non-exhaustive list of texts where Dickinson's appropriation of handwriting technology is presented in more detail includes: *Dickinson Unbound: Paper, Process, Poetics*, by Alexandra Socarides; *Black Riders: The Visible Language of Modernism*, by Jerome McGann; *Rowing in Eden: Rereading Emily Dickinson*, by Martha Nell Smith; and *Emily Dickinson's Open Folios: Scenes of Reading, Surfaces, of Writing*, by Marta L. Werner. In addition, the digital article "Dickinson: Cartoonist" (http://archive.emilydickinson.org/cartoon/index.html) and "Ravished Slates," a digital exhibition (http://www.emilydickinson.org/ravished-slates-re-visioning-the-lord-letters) – both available in the *Dickinson Electronic Archives* (http://emilydickinson.org) – provide highly visual

In “While it is alive,” Dickinson’s first stanza includes a “while” loop with multiple conditionals. The loop successfully processes when the following two conditions are met: 1) love (the antecedent of “it” as revealed by the second stanza) must be alive; and, 2) the poem’s speaker must be able to feel that love. With those conditions met, the speaker can recognize how the act of separation is both an act of splitting and an act of preparing one for the likeliness of separation (presumably as a result of death). The second stanza then elaborates on the reason why the conditional must be met in order for the speaker to see the redeeming qualities of division. The speaker must have love, the poem explains, in order to appreciate how, during the division caused by death, love prepares people for eternal life.

The conditional nature of the “while” syntax described above is likely available to most readers in a standard, descriptive interpretation of the poem’s text. However, by also appending the programmatic logic of how such “while” loops are utilized in computer programming changes the poem from a descriptive, one-time conditional check of love being alive into a continuous process of checking and re-checking for love’s presence. The “while loop,” as the WordPress Loop example tells us, will run until the conditional is no longer met. Applied to Dickinson’s poem, her “while loop” syntax encourages readers to recognize that the poem represents a continuously running function, and the attainment of eternal life is not a one-time pursuit, but instead, it is a continuous pursuit being repeatedly evaluated and re-evaluated

---

examinations of some of the many different ways in which Dickinson deployed the standard technology of the text-filled page beyond a typical “left-to-right, top-to-bottom” structure.

throughout one's entire life. The presence of love can help death be evaluated positively at first, but if that love is ever lost, meaning as soon as the conditional of having love is not met, death no longer becomes a preparation for eternal life.

By employing the "while" loop, Dickinson has made the poem, in a programmatic sense, dynamic. Instead of being a poem about the importance of having love in order to achieve eternal salvation, the poem is about the importance of maintaining love. Because of this emphasis on maintaining love, instead of being a descriptive poem, "While it is alive" is an instructive poem. It instructs processing entities (in this case, readers) to maintain love in order to maintain the possibility of eternal salvation.

This code-augmented distinction of reading Dickinson's poems as instructive technologies, as opposed to descriptive technologies, is more than just a pedantic exercise in reaffirming some nebulous sense of poetry's infinite interpretability. It demonstrates a way in which Dickinson's poetry is a didactic technology storing and disseminating shared cultural knowledge across generations. The instructive properties of the poetic syntax, enabled by programmatic elements such as "while" loops, allows readers to iterate through and react to information in much the same way the WordPress loop iterates across information extracted from a website's database. Because of this programmatic structure, instead of operating only as a passive text, the poem also operates as a programmatic script that gets processed by its reader and produces dynamic results. This programmatic structure within the poem

facilitates textual processing which may be helping Dickinson overcome the limitations of the text-based information technologies available.

The limitations of text-based technologies, as Sandra Gilbert points out, presented a challenge Dickinson constantly battled. Gilbert suggests that:

> Perhaps we should conclude that Dickinson spoke *English as a second language* and was our country's first great ESL poet. Remember how she complained that "They shut me up in Prose -- / As when a little Girl / They put me in the Closet -- / Because they liked me 'still' –"? What she really meant may have been that they shut her up in *language*, in the English language, the essential medium of our poetry, and she had to wrestle to bend it to her own purposes, to her will, not just for freedom but for accuracy, for absolute precision.[5]

If Dickinson is, indeed, wrestling and bending poetry to her own purposes, then, as the previous example begins to show, that bending seems to utilize programmatic and functional processing logic. One example of that functional processing logic is the previously described "while loop," but other types of programmatic loops are also common in Dickinson's poetry, and their presence reaffirms the value she seems to have found in leveraging poetic code as a means for working toward Gilbert's concept of "absolute precision."

[5] Gilbert, Sandra. "'If a lion could talk…': Dickinson Translated." *The Emily Dickinson Journal*, Vol. 2, No. 2 (Fall, 1993): 2.

For example, in the poem “For each extatic instant,” Dickinson uses a structure that, in coding terms, is known as a “for each” loop. She writes:

> For each extatic instant
> We must an anguish pay
> In keen and quivering ration
> To the extasy –
>
> For each beloved hour
> Sharp pittances of Years –
> Bitter contested farthings –
> And Coffers heaped with tears!
> (FP109)

“For each” loops are used in programming to iterate through a collection of items with no count for how large that collection might be. In contrast with the “for” loop, which is used to instruct an application to run the specific functions within the loop a pre-determined number of times, “for each” loops are used in cases where the number of possible instances are non-standardized. In Dickinson’s poem, for example, the number of “extatic instants” and “beloved hours” a person can have is not standard, so the poem relies on the “for each” loop in order to iterate across every possible moment and apply the necessary functions. In the case of an “extatic instant,” according to the logic of the poem, some amount of anguish must be paid in ratio to the amount of the “extasy” enjoyed. In the case of the “beloved hours”

loop, the ratios are made more explicit, with the cost being “sharp pittances of years,” “bitter contested farthings, and “coffers heaped with tears.”

Although, in order to avoid the confusion it might cause, I have avoided including code as a tool for supplementing my explanations, translating Dickinson’s poetry into another coding language might help demonstrate Dickinson’s coded syntax and how that syntax encourages not just the reading of a poem, but the processing of a poem. In the specific case of “For each extatic instant,” her syntax is so similar to a programmatic software function that her poem is easily translated into other programming languages. As an example of the opportunities for translation, I have translated Dickinson into PHP, the ‘poetic’ language of WordPress, as follows:

```php
<?php

    /**
     * Calculate amount of anguish a person has faced in life as a function of
     * the amount of 'extasy' the person has experienced.
     *
     * Author: Emily Dickinson
     * Translated to PHP By: Aaron Dinin
     * Date: 1/26/2013
     *
     * @param obj
     * @return obj
     *
     **/

    function calculate_extasy_cost($extasies)
    {
        $COE = array(                   // set a person's base level Cost of Extasy
                        'anguish'                       => 0,
                        'sharp_pitances_of_years'       => 0,
                        'bitter_contested_farthings'    => 0,
                        'tears_in_coffer'               => 0,
                    );

            foreach($extasies as $extasy)       // loop through extasies to determine kind and cost
            {
                if($extasy->type === 'extatic instant')          // if extasy is an extatic instant
                {
                    $extasy_cost = 1 / ($extasy->value);              // set cost as ratio
                    $COE['anguish'] = $COE['anguish'] + $extasy_cost; // add to anguish
                }
                else if($extasy->type === 'beloved hour')        // if extasy is a beloved hour
                {
                    // add costs in each category as appropriate
                    $COE['sharp_pitances_of_years'] = $COE['sharp_pitances_of_years'] + 1;
                    $COE['bitter_contested_farthings'] = $COE['bitter_contested_farthings']+1;
                    $COE['tears_in_coffer'] = $COE['tears_in_coffer'] + 10; // Avg. 10 per cry
                }
            }

        return $COE;    // Send Cost of Ecstasy object back to calling function
    }

?>
```

Figure 8: Emily Dickinson's "For each ecstatic instant" translated into PHP.[6]

The translatability of "For each ecstatic instant" directly into a programming language demonstrates how the parallels between Dickinson's poetic code and computer code exist beyond syntax. The conceptual structures themselves are

[6] Programmers often embed notes inside their code in order to explain what certain pieces of the code mean or do. Different languages use different symbols to represent notes, and when the processing machine reaches those symbols, it ignores any associated text. Since the code snippet shown here is written in PHP, the notes are delineated in two ways. The longer authorship note at the top of the snippet is set off from the rest of the code by /* and */. Shorter notes within the body of the code are set off using the // symbol.

just as translatable as the language, suggesting that they share more than just diction – they also share functionality.

As with the "while" loop, the functionality of the "for each" loop in "For each ecstatic instant" instructs continuous iteration. Readers should not only be aware that happiness has a cost, they should also be aware that each moment of happiness has an iterative cost constantly being recompiled and appended to the previous cost. Through that function of iteration and compilation, increasing happiness also increases joy's unavoidable associated burdens along some sort of pre-determined proportional relationship. The specific proportion, in this case, is not as important as the programmatic structure that again helps the static text of the page become a dynamic technology for processing information.

Beyond the looping functionality of "For each ecstatic instant," the poem demonstrates another programmatic concept in Dickinson's poetic structure: functions.[7] Functions are sequences of code that can be triggered to process data sent to them from other parts of a codebase. For example, WordPress has a function for determining the number of reader comments on a given post called `get_comments_number($post_id)`. The function can be called from within the WordPress Loop by sending to the function the post's unique identification number (the `$post_id`). The `get_comments_number()` function then searches the database to count the number of associated comments. If the function counts one or more

[7] Different programming languages refer to the concept of functions by alternate names, including: procedures, routines, and methods.

comments, those comments can be displayed beneath the post, with the visual representation of each comment also being processed by yet another function and loop.

Because “For each ecstatic instant” is not just a loop, but also a function for processing moments of “extasy,” every time such a moment appears within the Dickinson corpus, its cost is calculatable – in accordance with Dickinson’s poetic accounting practice – by running it through the “For each ecstatic instant” function. This ability to process poetic information from one poem through the functions of another poem allows for a reading of Dickinson’s poetry as a unified application as opposed to individual poems. As the following examples demonstrate, using this unified application approach to Dickinson’s poetry introduces interpretive options not immediately available within the poem’s text.

One example of the “For each ecstatic instant” function enhancing the reading of another Dickinson poem is the well-known poem “Success is counted sweetest”:

Success is counted sweetest
By those who ne’er succeed.
To comprehend a nectar
Requires sorest need.

Not one of all the purple Host
Who took the Flag today
Can tell the definition

So clear of Victory

As he defeated – dying –
On whose forbidden ear
The distant strains of triumph
Burst agonized and clear!
(FP112)

According to the direct text of the poem, the reason success is most appreciated by people unable to succeed is because understanding success "Requires sorest need" (4). However, the next stanza introduces a second reason why those unable to succeed most appreciate success, but the reason is not linguistically available without using Dickinson's "For each ecstatic instant" function. By plugging the presumed "extasy" of the second stanza's successfully taking the flag of victory into the "For each ecstatic instant" function, readers also know that people who achieve success "must an anguish pay." That anguish is the unique byproduct of experiencing an "ecstatic instant," and because it is not experienced without success, people unable to achieve success are likely to count success as being even sweeter than those who do achieve it. Hence, a "success is counted sweetest" by those who never experience the problems associated with success.

In another example of a programmatically enabled reading of Dickinson's poetry as a unified application, the "For each ecstatic instant"

function can be used to balance the payment metaphor of "I came to buy a smile – today –." Dickinson writes:

I came to buy a smile – today –
But just a single smile –
The smallest one opon your
dark – face
Will suit me just as well –
The one that no one else
would miss
It shone so very small –
I'm pleading at the
counter – sir –
Could you afford to sell?

I've Diamonds – on my fingers!
You know what Diamonds – are!
I've Rubies – like the Evening Blood –
And topaz – like the star!
'Twould be a bargain for
a Jew!
Say? May I have it – Sir?
(FP258)

Smiles, as physiological muscle movements, have no tangible costs. Despite their lack of tangible costs, a smile is capable of transferring sociological value. The value is a function of what the smile is representing: joy, happiness, excitement, satisfaction, love, and so on. But deriving value without incurring cost would violate one of the most fundamental philosophical (and physical) principles of the universe: *ex nihilo nihil fit* – nothing comes from nothing.[8] Everything in the universe must be derived from some other already-existent thing, and if a smile transfers sociological value, then that smile must have a calculable cost. In Dickinson's poetic universe, readers can calculate that cost through the "For each ecstatic instant" function. With the help of her function, Dickinson's programmatic poems turn the concept of the smile into an item with a calculable cost. As an item with a calculable cost, theoretically the receiving of a smile need no longer be based on solely emotional constructs such as making people happy, making people laugh, or making people love. Dickinson's poem wonders if the emotion of a smile can be purchased as though it were any other common commodity since the cost of the smile can be calculated – via the

[8] The concept dates back as far as Parmenides, a Greek philosopher living around the 5th Century BC. It is still a foundational principle in modern physics, and it is often referred to as the conservation of energy or the first law of thermodynamics. The first law of thermodynamics was originally articulated by German physicist Rudolf Clausius in 1850, roughly the same time period in which Dickinson is writing. While the connection between "I came to buy a smile – today –" and "For each ecstatic instant" is not definitive proof that Dickinson was directly referencing thermodynamic theory, the temporal symmetry is worth some attention, and its influence on Dickinson, directly or indirectly, seems likely. "Intellectuals of Dickinson's generation (born in the 1830s and early 40s) were compelled to re-think the nature and sources of order," Susan Howe points out. "During the second half of the nineteenth century, chemistry, poetry, folklore, philology, astronomy, biology, statistics, psychopathology, and psychical research met and mixed. There was a general feeling that truths of all sorts could be discovered by the processes of experiment, and among the truths that could be found by experiment, and the chance involved in experiment, were spiritual, metaphysical, and moral ones" ("Experience is the Angled Road," 34-36).

`ForEachEcstaticInstant()` function – as though it were any other common commodity.

The two examples offered here of processing information in one poem by referencing a sort of programmatic function created by another poem demonstrates an available reading of Dickinson's poems, but the reading is not a required reading or the only possible reading. In this regard, we can make an important distinction between code and poetry. Computer code uses a concept called "namespacing" which enforces a processing path that, unlike Dickinson's poetry, is not optional. Among other things, namespacing prevents two functions from having the same name and forces the calling of a function from one part of a program to execute to completion the specific function being called. If, for example, a function within a program was actually named `ForEachEcstaticInstant()` and that function was called elsewhere, the interpretive pathway is clear. In contrast, the "ecstatic instants" throughout Dickinson's poetry are not so rigidly defined. The reader has the option to connect to the `ForEachEcstaticInstant()` poem and use it as a processing function, as I did, but the connection is not required nor is it specifically directed by the technology.

Differences like these are examples of my analogy between code and poetry breaking down, but in a productive way. Because, by their nature, all analogies must break down at some point of deconstruction, these moments of breakage contain instructive information about the unique qualities of an analogy's binary components. In this case, because the analogy between code and poetry has been created as the product of my introducing two questions –

“is code poetry?” and “is poetry code?” – the breakdown of the analogy begins to offer potential answers.

In the case of the first question spurred by the WordPress slogan – “Code is Poetry” – if your understanding of poetry is one that requires flexible interpretability, then the strict processing structure enforced by computers in order to operate code suggests that no, code cannot be poetry. However, regarding the second question – “is poetry code?” – the breakdown of the analogy introduces the question of who or what is the interpretive agent. If the interpretive agent is a computer, then the strict interpretation principles of digital processors prevent flexibility. However, when the interpretive agent is a person, the required structure of the information is less strict because the human mind is a more flexible processing machine.

The concepts of loops and functions are just two of the many programming conceits present in Dickinson’s poetry. A brief return to the WordPress Loop provides a helpful introduction to another often-used programmatic concept: conditional statements. The previously discussed “while” loop is an example of a conditional statement being evaluated multiple times. However, most conditionals in programming are only evaluated once, creating a concept known as an “if-then” conditional. Unlike the “while” loop conditional, which constantly evaluates the validity of a statement and then directs a response accordingly, an “if-then” conditional checks the validity of a claim once and only once and then processes the code nested inside the bounds of the condition if and only if the conditional

validates as true. In the case of the WordPress Loop, the program calls an "if-then" conditional during its rendering process immediately prior to the Loop, and it uses the conditional to evaluate whether or not the Loop should be run at all. The conditional makes this determination by checking for the existence of any blog posts before even attempting to loop through and display the posts, with the programmatic logic being that, in the event no posts match the criteria determined by the user, the user should be notified. The notification of no matching criteria appears as the output of a third, optional syntactic unit in an "if-then" conditional: an "else" clause. An "else" clause in a conditional provides instructions for a program to execute only if a conditional does not validate as true. In the case of WordPress, the "else" clause tells the website to show a message alerting the user that no matching posts exist. A natural language transcription of the conditional logic for the WordPress Loop's "if-then-else" construction described above could look like the following:

> **"If" Clause:**
>
> If any blog posts meet the conditions defined by the current user of this website, such as desired post date, desired post subject, or desired post author…
>
> **"Then" Clause:**
>
> Then run the WordPress Loop and display all the appropriate posts…

"Else" Clause:

Otherwise ("else"), alert the user that no posts exist matching his/her desired criteria.

Conditional "if-then" and "if-then-else" clauses are core patterns within interactive programs. Most programming languages include some version of conditional statements, and the pattern can appear thousands of times within a single piece of software. Because of the commonness of these conditional structures in software, and because of their importance in managing the logic that enables most programs to respond to user and environmental input, if Emily Dickinson's poetry is code, then one clue to such a reading might be the prevalence of similar conditional logic patterns.

Even a brief glance at first lines of Emily Dickinson's poems reveals that "If-then" conditional structures are prominent. Twenty-nine poems begin with some form of an "if-then" conditional, and many other poems incorporate conditionals at later points. These conditionals often create the same kinds of interactive logical progressions as the conditional statements used in computer programming logic. One example appears in "If I can stop one Heart from breaking." Dickinson writes:

If I can stop one
Heart from breaking
I shall not live
in vain
If I can ease one

Life the Aching

Or cool one Pain

Or help one fainting

Robin

Unto his Nest again

I shall not live

in vain.

(FP982)

The first conditional is a simple, single-factor conditional. If the speaker can stop one heart from breaking, then the conditional validates as true. By validating as true, meaning if the poetic speaker successfully stops a heart from breaking, then the result is that her life has some benefit to, presumably, humanity.

The second conditional in "If I can stop one Heart from breaking" is a more complex form of the conditional statement structure because it has multiple possible opportunities to validate as true: either the speaker can ease someone's life of aching, or the speaker can dull a pain of someone (or possibly something), or the speaker can help a robin back into its nest. If any of those three clauses of the conditional validate as true, then the speaker will not, as she perceives it, be living in vain. This kind of multi-clause conditional is also commonly used in coding, often even using the same "or" syntax to represent multiple possible ways by which a function can validate as true.

The prevalence of conditional statements in Dickinson's poem is enough to confirm an underlying programmatic structure, but where the programmatic relationship between code and poetry begins to produce interpretive value is in the ability to analyze the poetry as though it were code. Programmers, like poets, have many options that output the same functional results. Analyzing code, as a result, and also like analyzing poetry, often includes a process of considering the syntactic options the author did not take but could have.

In programmatic structures, when the author decides to use an "or" logic operator, she is making a decision based on the need to account for multiple validation conditions. Those multiple validation conditions have a relationship to one another, and in the case of an "or" clause, the relationship is that at least one condition within a group of conditions must be true. The primary alternate to this kind of condition is to use an "and" clause. Using an "and" clause requires that both relationships within a condition must be true in order to validate as true. A simple example of considering both possible relationships is to replace the "or" logic operators in the poem's second "if-then" clause with "and" logic operators. Using the alternate operators, the second clause would read as follows:

If I can ease one
Life the Aching
And cool one Pain

And help one fainting

Robin
Unto his Nest again
I shall not live
in vain.

Using an “and” operator instead of an “or” operator dramatically alters the meaning of the poem. The speaker must accomplish all three tasks – ease one life from aching *and* cool one pain *and* help a fallen robin back into his nest – in order for her life to not be in vain.

From a programmatic perspective, Dickinson’s use of an “or” conditional in “If I can stop one heart from breaking” is notable not solely because of her choice to use “or” instead of “and” in the second conditional, but for her choice to use the “or” conditional to validate three alternate possible conditionals for a programmatic outcome identical to the outcome of the first conditional. If Dickinson had employed “and” operators in the second conditional, the poem would be equating two possible conditionals as equal to each other: the single act of stopping one heart from breaking would have the same result as easing aching, cooling pain, and helping a fallen bird. However, as written, if Dickinson’s poem were to be read as a program, it would be considered sloppy (i.e. inefficient) code. The optimal poem should be:

If I can stop one
Heart from breaking
Or ease one
Life the Aching

Or cool one Pain
Or help one fainting
Robin
Unto his Nest again
I shall not live
in vain.

And yet, that is not the poem. Instead, insisting on efficient use of programmatic logic in Dickinson's poem enforces a reading of "If I can stop one Heart from breaking" that makes "not living in vain," for Dickinson, a terminating event. Programmatically, such an event would be an event that, whatever its processes, prevents the program from continuing to the next line either through returning some sort of value or exiting the program and returning an error message. In the case of Dickinson's poem, if the reader interprets "not living in vain" as a terminating event, then once the speaker stops one heart from breaking, she never needs to run the second conditional. This termination gives precedence to the first conditional as a more important conditional within the program's logic.

An interpretive reading of the poem reiterates the above-described programmatic logic. The second, longer conditional reads as though the speaker is introducing desperation. The speaker's priority is to stop one heart from breaking, but, if for some reason she is unable to accomplish that – though hopefully she can – then she would accept any of the other three possibilities as less-desirable but still viable alternatives.

The hierarchical distinction of conditional logic operators in Dickinson's poetic structure is worth considering because of the indirect interpretation offered by the missing "else" clause. As is the case with any conditional in programmatic logic, if a condition is not met, the code within the conditional is skipped, and the program automatically moves to the lines immediately following the constraints of the conditional as though no code within the conditional exists. In the cases of "If I can stop one Heart from breaking," the separation of the first and second conditionals creates desperation because the missing "else" clause implies no alternative if no condition is met. What happens if the speaker cannot stop a heart from breaking, ease a life of aching, cool a pain, or help a robin back into its nest? The poem makes no attempt at an answer, and its lack of an answer offers the poem's most troubling conceit. If the conditional logic was completed with an "else" clause, and none of the other conditions were met, then the poet would force an answer that, by its programmatic nature, would be all-encompassing. Instead, the nothingness caused by the possible lack of a terminating event – the lack of an "else" clause – makes Dickinson's use of split conditionals not sloppy coding, but a structure capable of heightening the poem's sense of desperation. By splitting her conditionals and not closing the poem's functional logic, Dickinson programmatically creates a powerful aposiopesis perhaps more powerful in its unfinished silence than any concluding stanza could ever be.

In one of Dickinson's other deployments of the "if-then" construct – "If you were coming in the Fall" – the poet applies another common coding

practice when using conditionals known as an “else if” clause. An “else if” clause creates multiple programmatic pathways for the evaluation of the same object by re-evaluating the object against different possible logic checks. Doing so lets the programmer define a variety of responses to the same action. In the poem, Dickinson writes:

> If you were coming in the Fall,
> I’d brush the Summer by
> With half a smile, and
> half a spurn,
> As Housewives do, a fly.
>
> If I could see you in a
> year
> I’d wind the months in balls –
> And put them each in
> separate Drawers,
> For fear the numbers fuse –
>
> If only Centuries, delayed,
> I’d count them on my Hand,
> Subtracting, till my fingers
> dropped
> Into Van Dieman’s Land.

If certain, when this life was out –
That your's and mine, should be –
I'd toss it yonder, like a
Rind,
And take Eternity –

But, now, uncertain of the
Length
Of this, that is between,
It goads me, like the
Goblin Bee –
That will not state –
It's sting.
(FP356)

Each of the first four stanzas relies on an "if-then" conditional that responds to the same action. By offering different responses based on the different possible versions of the inputted action, the conditionals are, despite not specifically using "else if" syntax, functioning as "else if" clauses. If any of the clauses were to evaluate as true, the speaker would avoid the outcome of the "else" clause outcome of the fifth and final stanza, but the way of avoiding the final "else" clause differs based on the different conditional statements. If the unnamed person would be arriving in the fall, the speaker would be happily speeding through her experience of the summer in order to hasten fall's

arrival (stanza 1). If the mysterious person would be arriving within the year, the speaker would have to make a conscious effort to separate the months leading to his or her arrival so that the months would not all run together and be easily overlooked (stanza 2). If the person would promise to arrive within centuries, the speaker would happily wait until death or beyond (stanza 3). And, as the fourth stanza explains, if the speaker were to know that death would bring the absent person to her, she would welcome even that terminal fate. But, as explained in the poem's "else" clause (stanza 5), in the event that none of the prior conditions are met (and while none of those conditionals are met), the speaker's only option is to be frustrated by an uncertain future.

Notice how the terminal "else" clause in "If you were coming in the Fall" produces a definite outcome that was completely lacking in "If I can stop one Heart from breaking." If analyzed as functions, whenever the "If you were coming in the Fall" function is run, it will always produce a definitive result, despite the information in the data being processed. In contrast, if none of the conditions are met when running the "If I can stop one Heart from breaking" function, nothing happens – the poem returns emptiness.

While the syntax parallels between poetic and programmatic conditionals encourage the kind of expanded interpretive reading noted above, the value of introducing the parallels extends, like the loop, beyond interpretation and into technological functionality. In the case of conditionals, the programmatic syntax helps the poems establish poetic movement. Or, in programmatic terms, the conditionals help make Dickinson's software

dynamic, which, in turn, helps Dickinson's poetry overcome the limitations of her English-language-based, static text technology of information storage and dissemination.

This ability to create poetry dynamically adaptable poetry through the use of programmatic language may be what, for Dickinson, makes poetry a more descriptive, vibrant, and successful medium for storing and transmitting information than other artistic media. Not only is it a claim she makes surreptitiously by employing syntax, like loops and conditional logic, that forces conceptual processing, it is a claim she overtly makes in her poetry. One such place where this claim appears is in "How the old Mountains drip with Sunset," where Dickinson writes:

> How the old Mountains drip
> with Sunset
> How the Hemlocks burn –
> How the Dun Brake is draped
> in Cinder
> By the Wizard Sun –
>
> How the old Steeples hand
> the Scarlet
> Till the Ball is full –
> Have I the lip of the Flamingo
> That I dare to tell?

Then, how the Fire ebbs like Billows –

Touching all the Grass

With a departing – Sapphire – feature –

As a Duchess passed –

How a small Dusk crawls

on the Village

Till the Houses blot

And the odd Flambeau, no men

carry

Glimmer on the Street –

How it is Night – in Next

and Kennel –

And where was the Wood –

Just a Dome of Abyss is

Bowing

Into Solitude –

These are the Visions flitted

Guido –

Titian – never told –

Domenichino dropped his pencil –

Paralyzed, with Gold –

(FP327)

Contrasting my code-based poetic analysis of "How the old Mountains drip with Sunset" with Cristanne Miller's traditional interpretive analysis of the same poem provides an instructive example of how Dickinson could be viewing poetry not necessarily as a medium for generating aesthetic beauty, but as a technology for storing and transmitting cultural data. Miller interprets Dickinson's poem as demonstrating how poetry is a more vibrant art form than painting, explaining that the poem uses "competitively fanciful description, where the speaker flaunts verbal skill surpassing the representational art of Italian masters."[9] However, whereas Miller attributes the poetry's ability to "describe something beyond the ability of the greatest painters [as being] reflected in the poem's competing rhythms, emphatic line beginnings, and uneven line lengths," I wonder if Dickinson creates a contrast with the Italian masters by acknowledging the way in which poetry is processed by its readers. Although Dickinson would not have the vocabulary to consider poetry a form of digital code, she does appear to approach the human mind as a sort of poetry processing machine that responds to poetry in programmatic ways. That approach appears in "How the old Mountains drip with Sunset" through the contrast she creates between poetry and painting.

Painters provide visual images directly to their audiences. As a result, the images audience members experience are consistent across viewers. In the

[9] Miller, Cristanne. *Reading In Time: Emily Dickinson in the 19th Century*. Amherst, MA: University of Massachusetts Press, 2012. 61-62.

case of poetry, poets have no fixed visual points around which to center images. Instead, poets must rely on language to evoke meaning through each audience member's individual experience and perception of every word.

In a medium like poetry, where its core components – words – are naturally inconsistent across readers (and even internally inconsistent for the same reader), creating static, consistent images is impossible. However, during the historical period when Dickinson is writing, the technology to generate, store, and transmit moving images does not exist. As a result, while words might not be as successful as paintings in their ability to maintain consistent visual output, the mutability of language would make poetry a powerful medium for creating moving images.

Without technologies like television and video, the creation of moving images takes place in people's minds. In order to create those images, Dickinson has to, in essence, program the mind to see what she wants it to see. In "How the old Mountains drip with Sunset," Dickinson explains the concept of programming people's minds by contrasting poetry with painting through the use of the word "how." Removing every instance of the word "how" in "How the old Mountains drip with Sunset" still results is a perfectly readable poem:

The old Mountains drip
with Sunset
The Hemlocks burn –
The Dun Brake is draped
in Cinder

By the Wizard Sun –

The old Steeples hand
the Scarlet
Till the Ball is full –
Have I the lip of the Flamingo
That I dare to tell?

Then, the Fire ebbs like Billows –
Touching all the Grass
With a departing – Sapphire – feature –
As a Duchess passed –

A small Dusk crawls
on the Village
Till the Houses blot
And the odd Flambeau, no men
carry
Glimmer on the Street –

It is Night – in Next
and Kennel –
And where was the Wood –
Just a Dome of Abyss is

Bowing

Into Solitude –

These are the Visions flitted

Guido –

Titian – never told –

Domenichino dropped his pencil –

Paralyzed, with Gold –

Because the word "how" is non-essential to the content and structure of the poem, its continuous presence suggests an explicit choice by Dickinson. The word "how" changes the meaning of every image it modifies by deemphasizing the image and putting focus on the specific way in which each image is rendered by a reader. For Dickinson, creating dynamic and moving images (something a single painting cannot accomplish) appears to be possible by relying on *how* readers derive mental images based on the words they read. Thus, a poet, in comparison to a painter, can deploy language capable of being processed by a reader in order to produce a dynamic mental image.

Authors in the digital world call that task of creating language that gets processed as instructions for producing dynamic visual objects the process of "writing code." The primary difference between writing poetry and writing code seems to be whether the machines processing the code have interpretive consistency. In the case of computer code, the machines are designed to

interpret the code the same way every time. In the case of poetic code, every reading machine – every human being – interprets poetic code differently.

While infinite and evolving interpretability might be an artistic benefit of literary texts (a benefit considered and analyzed in my fourth chapter), as a developer trying to produce language executed consistently across multiple machines, the variable interpretability of language appears to be one of Dickinson's biggest frustrations. She alludes to her frustrations when, in a response to a letter from one of her primary correspondents, Thomas Wentworth Higginson, Dickinson writes: "I almost inferred from your accent you might come to Amherst. I would like to make no mistake in a presumption so precious – but a Pen has so many inflections and a Voice but one, will you think it obtuse, if I ask if I quite understood you?"[10] If the "accent" – or rather, lack of accent – in written language troubles Dickinson in her direct correspondence, a medium like poetry likely magnifies the problem because she cannot expect to always have intimate knowledge of the reader and his or her perceptions of text. Perhaps this lacking familiarity is even one of the reasons why Dickinson would elect to not mass publish her poetry. Instead, she distributed her poems primarily to people she knew, limiting the comprehension of her written accent to people with some knowledge of her linguistic tendencies.

Dickinson's frustration with the variable interpretability of language is so prominent that much of the fascicle in which "How the old mountains drip

[10] Letter to Thomas Wentworth Higginson, 1876 (L470).

with sunset," Fascicle 13, alternates between the same two themes featured prominently in the poem itself: trying to describe a sunset and trying to understand how people process information.[11] For example, the third poem of Fascicle 13 opens with the lines:

> You see I cannot see – your
>
> lifetime –
>
> I must guess
>
> (FP313)

In this poem Dickinson emphasizes her struggle with the inability of being able to comprehensively understand the perspectives of people whose lives had been filled with different experiences. Since different experiences over a lifetime lead to different interpretations of language, Dickinson realizes she is trapped in a medium that, by it's very nature, cannot produce consistently interpreted texts.

As Dickinson's poems progress through the fascicle, she continues struggling both with how people process information differently based on experiences, and how people's processing of information, as opposed to the language itself, is what creates meaning. For example, in "To die – takes just a

[11] Socarides warns that, "While it may be tempting to read all of the writings in one fascicle as related to each other, as most critics of the fascicles have done, comparative archival research and close attention to compositional method suggest that connections *across* sheets are not Dickinson's primary concern" (35). But Socarides does note that, even if relating all writings in a fascicle to each other is not Dickinson's primary concern, "Taking the fascicle sheet as Dickinson's primary unit of construction reveals how Dickinson grouped poems together and allowed them to speak back and forth to each other in non-uniform ways – in effect, how Dickinson used this material unit to register her own resistance to both the static lyric moment and an all-encompassing narrative." (36). My choice of reading the poems of Fascicle 13 in conversation with one another is not to impose an all-encompassing narrative, as Socarides warns against. Instead, the programmatic comparisons within my reading are meant to consider how the grouped poems are speaking back and forth to each other.

little while –," Dickinson describes how the physical act of dying involves very little pain. Instead, the poet explains in the final stanza that:

> The absent – mystic –creature –
> That but for love of us –
> Had gone to sleep – that
> soundest time –
> Without the weariness –
> (FP315)

The pain of death, Dickinson reasons, is created by the living people remembering their own experiences with and feelings for the deceased. As Dickinson writes in the third line of the poem, the act of dying "doesn't hurt," and therefore, the painful experience of death most people perceive is not the experience as felt by someone who is dying. The pain of death is the pain of the survivor's perspective: The pain is experiencing the death of someone you love.

Within the context of a fascicle of poems analyzing the ways in which human responses and perspectives to the same events can vary, Dickinson includes no less than five poems attempting to describe a sunset.[12] Since the final of those five sunset poems is the previously discussed "How the old Mountains drip with Sunset," a poem in which Dickinson specifically articulates the importance of considering the ways in which people experience

[12] A result of the interpretability of language with which Dickinson struggles is that, if read in certain ways, at least four additional poems in Fascicle 13 could be interpreted as being partially related to describing a sunset: "Of Bronze – and Blaze –" (FP319); "Put up my lute!" (FP324); "The lonesome for they know not What –" (FP326); and "Of Tribulation – these are they" (FP328).

and interpret events, the presence of so many poems about the same topic positioned so near one-another suggests that the sunset poems are examples of Dickinson's struggles with the interpretive inconsistencies of language. The multiple sunset poems in Fascicle 13 represent Dickinson exploring alternate poetic strategies for rendering the same event.

Dickinson's varying treatments of sunsets begin with the poem "She sweeps with many-colored Brooms," a poem in which Dickinson personifies the sun as a cleaning housewife. Dickinson writes:

She sweeps with many-colored Brooms –
And leaves the shreds behind –
Oh Housewife in the Evening West –
Come back – and dust the pond!

You dropped a Purple Ravelling in –
You dropped an Amber Thread –
And now you've littered all the East
With Duds of Emerald!

And still, she plies her spotted
Brooms,
And still the Aprons fly,
Till Brooms fade softly into stars –
And then I come away.
(FP318)

This poem describes a sunset from the perspective of a speaker watching the event near a pond in the midst of a clearing within a forest. The pun on the word "leaves" in the second line evokes the image of the setting sun's colored rays coming through the forest trees, with the pond not being shadowed because the trees are too far away. However, this vivid image of the sunset quickly becomes a dramatic example of alternate perspectives when we contrast it with Fascicle 13's final sunset poem – the previously discussed "How the old Mountains drip with Sunset." In that final sunset poem, the fifth stanza describes a forest as "Just a Dome of Abyss" (25). However, in the fascicle's first sunset poem, the perspective is written from inside that "Dome of Abyss" – a spot where the forest dome looks less dark and solid, and the sunset looks more vibrant.

Fascicle 13's second sunset poem is more macabre. In contrast to the oft-romanticized beauty of sunsets, Dickinson describes a sunset's ability to foreshadow, both physically and metaphorically, total darkness, explaining:

> There's a certain Slant of light,
> Winter Afternoons –
> That oppresses, like the Heft
> Of Cathedral Tunes –
>
> Heavenly Hurt, it gives us –
> We can find no scar,
> But internal difference –

Where the Meanings, are –

None may teach it Any –
'Tis the Seal Despair –
An imperial affliction
Sent us of the Air –

When it comes, the Landscape listens –
Shadows – hold their breath –
When it goes, 'tis like the Distance
On the look of Death.
(FP320)

The reference to "the Heft of Cathedral Tunes" in the poem's third and fourth lines is a link to the kind of shadowy darkness implied by sunsets. Just as church bells toll when announcing deaths and cast a gloom over their surroundings, some sunsets – particularly New England's cloudy, winter sunsets – can make an already cold and drab world even colder and more deathlike. Dickinson again seems to be linking the alternate perspective from the fifth and final sunset poem to this earlier one when, in line seven of "How the old Mountains drip with Sunset," she references the sun setting on the "old Steeples." The return reference to church steeples evokes the fascicle's earlier sunset poem while providing a subtle reminder that, even though some sunsets may be radiant, the weighty "Cathedral Tunes" are ever-present, and

death makes no distinction between the gorgeous days and the ones covered by clouds.

Fascicle 13's third sunset poem pushes questions of perspective beyond the poem itself by providing the following descriptive vision:

Blazing in Gold – and
Quenching – in Purple!
Leaping – like Leopards – to
the sky –
Then – at the feet of the
old Horizon –
Laying it's Spotted face to die!

Stooping as low as the
kitchen window –
touching the Roof –
And tinting the Barn –
Kissing it's Bonnet to
the Meadow –
And the Juggle of
Day – is gone!
(FP321)

As with the first and second sunset poems, this poem seems to be recalled by the final sunset poem when Dickinson writes, "How a small Dusk crawls / on

the Village / Till the Houses blot" (16-18). The village of the final poem evokes the "Roof" (10) and "Barn" (11) of the fascicle's third sunset poem, while the blotting of the houses in the fifth sunset poem parallels the third poem's spots of the leopard. In addition, Dickinson's opening description of the sunset as "Blazing in Gold" is directly echoed in the last line of the fifth sunset poem when Dickinson describes Domenichino, one of the poem's three famed Italian painters, as being "Paralyzed, with Gold –" (32). When read in concert with a poem where an Italian master painter is paralyzed by the sunset's gold, the unhesitating description of the golden sunset by Dickinson evokes questions of perspective even as they relate to artistic perception. Subjects feared in one artistic medium because of a technology's limitations are eagerly welcomed in other artistic mediums because of a technology's flexibility.

The fourth sunset poem of Fascicle 13 approaches a sunset from the perspective of the darkness in a sunset's immediate aftermath. The poem asks the night what happened to the sun:

> Good Night! Which put
> the Candle out?
> A jealous Zephyr – not a
> doubt –
> Ah, friend, you little knew
> How long at that Celestial
> wick
> The Angels – labored diligent –
> Extinguished – now – for you!

It might have been the
Light House Spark –
Some Sailor – rowing in the
Dark –
Had importuned to see!
It might have been the
Waning lamp
That lit the Drummer
from the Camp
To purer Reveille!
(FP322)

By considering the result of a sunset, as opposed to just the immediate beauty of the sunset itself, the poem introduces metaphysical questions of perspective. It reminds readers that, despite the beauty of sunsets, their results are darkness. When considered from all perspectives – not just the artists trying to recreate an image – the impending darkness of sunsets is more troubling than the sunsets are beautiful. Sailors, as Dickinson's poem notes, are less inclined to think of a sunset as beautiful, and, instead, they are likely to perceive sunsets as harbingers of navigational dangers. As with the other sunset poems, Dickinson also incorporates the fourth poem's alternate perspective on sunsets into the fascicle's fifth sunset poem when she writes, "How it is Night – in Nest / and Kennel" (22-23), referencing types of houses

perhaps not as fortunate as human houses in their ability to have light without the sun. In those houses, instead of representing beauty, a sunset likely represents darkness and, perhaps, an increased likelihood of unseen predators.

Following four other perspectives on sunsets, by the time readers reach the end of the fifth and final sunset poem, they should be willing to accept Dickinson's assertion that the greatest painters would have feared painting sunsets. Considering the variable perspectives from which Dickinson has already portrayed sunsets, how could a painting capture a sunset's complexities?

But Dickinson's poems are not denigrating only painting. Her need to describe a sunset across at least five different poems within a single fascicle is an indication not of poetry's ability to provide comprehensiveness where painting cannot, but is a further indication of the complexity of a world in which technological representation can only serve as a piecemeal surrogate for actual events. Poetry is, in this regard, a simulation. Its readers process it and, in so doing, the poems evoke the perception of, among other things, real-world events, objects, and emotions. But poems do not give readers direct access to those things. Instead, poetry appears even more similar to computer code because both code and poetry depend on uniquely-defined linguistic patterns to produce interpretable character strings that, when evaluated by a processing agent, generate simulated events.

This suggestion of poetry and code having similar structures and purposes, along with my previous examples, are further corroborated by

descriptions of Dickinson's poetry written by other readers. For example, Adrienne Rich describes Dickinson's poetry as "a language more varied, more compressed, more dense with implications, more complex of syntax, than any American poetic language to date."[13] Rich's description of Dickinson's poetry marks it specifically as a particular kind of poetic language, which makes it both uniquely identifiable for the syntax it uses to generate interpretable meaning, and it makes Dickinson's poetic language distinct from, say, Emerson's poetic language, T.S. Eliot's poetic language, Chaucer's poetic language, and any other identifiable poetic language ever written. Just as these qualities (i.e. unique and identifiable syntax structures as part of a text that generates interpretable meaning) characterize poetic languages for Rich, they characterize programming languages. Does it matter if one type of programming language is meant to be processed by human readers and the other type of programming language is meant to be processed by machine readers?

If Fascicle 13 is any indication, variability of the processing agent might be precisely what distinguishes poetic syntax from computer syntax and ultimately limits the analogy between code and poetry. On its surface, computer code has the distinct advantage of being processed by machines with fixed interpretive patterns. In contrast, and as Fascicle 13 reminds readers numerous times, the interpretive patterns of human readers are

[13] Rich, Adrienne. "Vesuvius at Home: The Power of Emily Dickinson (1975)." *On Lies, Secrets, and Silence*. New York: W.W. Norton & Company, 1979. 163.

anything but fixed.[14] However, despite their seeming consistency, machine-audiences for programming languages are, like poetic audiences, not fixed. In fact, many software engineers would eagerly agree with Dickinson's assessment that "You see I cannot see – your / lifetime – / I must guess" (1-3). Like Dickinson, software engineers are limited to the knowledge of their own systems. They develop software on their own machines, but thousands of different types of computers exist around the world, and even for computers that are ostensibly the same (i.e. Macs, PCs, iPhones), the different types of computers have millions of different combinations of processing power, screen resolutions, operating systems, hardware, software, and other settings that make creating complex code interpreted identically on every computer impossible. Just as Dickinson can send poems to her confidants to test their ability to properly evoke the desired meaning when read by alternate readers,[15] developers can test their programs on other computers to make sure they run properly in alternate environments. But, for both poetic code and computer code, once the code's creator distributes it beyond her controllable environments, she has no way of accounting for every possible "lifetime" of the processing agents.

---

[14] Even Dickinson struggles with her own interpretive patterns. Her relationship with the Bible, for example, is a complex relationship that elicits from Dickinson significant interpretive conflict. Alicia Ostriker describes Dickinson's personal struggles interpreting the Bible when she writes: "To read Dickinson on God (etcetera), then, is to divest oneself of the desire for a single 'correct' interpretation of scripture, to accept willy-nilly a plurality of interpretations, and to begin to realize that the Bible, like Emily herself, *asks* to be read like this, to be plurally and not singly interpreted. Putting this another way: to the powerless Dickinson daughter scripture is an antagonist, to the passionate woman Emily it is a lover, to the powerful poet Emily Dickinson it is a poem out of which one makes, of course, other poems" (Ostriker, 467).

[15] Good resources for understanding Dickinson's practice of sending poems to close confidants for feedback include *Open Me Carefully: Emily Dickinson's Intimate Letters to Susan Huntington Dickinson*, by Martha Nell Smith and Ellen Louise Hart, and *Emily Dickinson's Correspondences: A Born-Digital Textual Inquiry*, edited by Martha Nell Smith and Lara Vetter.

By removing any misconceptions that interpretive audiences for programming languages is standardized, so far as I can tell, the only real distinguishing factor keeping poetry from being code is that poetry's audiences are alive in ways that computers are not. If only one distinguishing feature is needed to erect a boundary between programming languages and poetic languages, perhaps the fact that humans and computers are not alive in the same kinds of ways is distinction enough.[16] But the problem with separating poetry from programming based on a distinction between audiences is that, while much of my discussion of programming has described a machine-level audience for code, not only does code have a human audience, in most cases, that human audience may be a more important audience when considering programmatic structures and syntax. In contrast to code's human readers, computers have no opinions on the structure of code. They have no opinions on things like syntax and logic. Most of the time, computers are not even reading code in the original programming languages in which programs were written because the majority of developers write code in what are called "high-level" programming languages that dramatically abstract the machine code syntax into human-readable syntax.[17] Thus, the

[16] Claiming that computers are not alive does leave room for counter-arguments involving the "singularity" – the theoretical moment when digital technologies advance enough to become self-aware – but, as Jaron Lanier warns, such arguments "echo Marxist social determinism and Freud's calculus of perversions. We rush ahead of skeptical, scientific inquiry at our peril, just like the Marxists and Freudians" (Location 436).

[17] "High-level" programming languages are intended to make code-based functionalities more easily readable and definable for human users, while "low-level" programming languages use syntax closer to machine-readable code that is largely incomprehensible when compared with standard human diction. "Low-level" programs generally run faster than "high-level" programs because they require little or no compiling or interpretation by the computers processing them, but they are usually harder to use because the syntactic requirements are more demanding for most developers.

readers of code are developers. They are people, and they approach software in much the same way an Emily Dickinson reader approaches an Emily Dickinson poem. At first glance, a coded text looks readable because it uses words and characters common in natural languages. But reading the text is difficult. The syntax and word order is different from traditional prose. The logic and organization sometimes requires specialized knowledge. But usually, if you read it enough times, and if you stare at it long enough, you can extract meaning.

This interpretive juncture – the point at which a person is extracting interpretable meaning from a language-encoded document – is the point at which poetry and code seem to most overlap one-another. It is also the most likely point at which one technology begins to be viewed, correctly or incorrectly, as the other. The reason would seem to be that it represents the moment the prioritizing audiences become the same, and, as a result, their needs become the same. When machines are processing machine code, the code does not need to be poetry because computers do not need poetry. But, when machine code is being processed by people, we can return to Muriel Rukeyser's explanation of poetry that opened this chapter and wonder if we should consider code processed by people as being poetry because, as Rukeyser asserts, "If there were no poetry on any day in the world, poetry would be invented that day." Code is not poetry in the sense that code and poetry are the same technologies, but perhaps the poetry of code is supplied by the human interpretive component that invents poetry when it is needed. Code becomes poetry when its users require a unique poetic syntax that

extends the code beyond the limitations of the technological medium in which it is contained just as Dickinson extends her writing beyond the limitations of the technology of poetry.

Conclusion: Metaphor as Technology

In a talk given at Amherst College in 1933, Robert Frost made the case that "Education by poetry is education by metaphor."[1] Frost's suggestion is a telling reference point for thinking about this project's general path. The project begins by introducing a software metaphor as an interpretive tool for a Walt Whitman poem and ends by speculating about the relationship between programmatic code and Emily Dickinson's poetry. In other words, *Hacking Literature* starts with metaphor, ends with poetry, and, in between, repeatedly asks and attempts to answer what makes technologies such as those in literature, and specifically the technology metaphor, capable not only of storing and disseminating information but of building knowledge – capable of, in essence, educating. In doing so *Hacking Literature* attempts to foster what Frost calls "education by metaphor."

By suggesting this project might be educating by metaphor, I do not mean the project is poetry, nor is Frost claiming that all education by metaphor is education by poetry. Frost's comments are about the value of learning comparatively, both through and by metaphor – a value which poetry instills, and a value this project holds in highest regard. As Frost writes:

> What I am pointing out is that unless you are at home in the metaphor, unless you have had your proper poetical education in the metaphor, you are not safe anywhere. Because you are not at ease with figurative values: you don't know the metaphor in

[1] Frost, Robert. "Education by Poetry." *Amherst Graduate's Quarterly*. Feb. 1931. Web. 17 Apr. 2014. http://moodyap.pbworks.com/f/frost.EducationByPoetry.pdf. 2.

> its strength and its weakness. You don't know how far you may expect to ride it and when it may break down with you. You are not safe with science; you are not safe in history.[2]

Metaphor, for Frost, and metaphor, as deployed in this project, has greatest value in the comparative thinking it demands. In contrast, if code and poetry were the same, and if digital technologies and literary technologies were the same, then users would not need to distinguish between them. Nor, I imagine, would they benefit from the kinds of critical comparisons this document recommends.

But code is not poetry. Nor, by the way, is poetry code, however coded it may be. And digital technologies, while capable of supporting many similar social and cultural goals, are not the same as literary technologies. "All metaphor breaks down somewhere," Frost explains. "That is the beauty of it. It is touch and go with the metaphor, and until you have lived with it long enough you don't know when it is going. You don't know how much you can get out of it and when it will cease to yield. It is a very living thing. It is as life itself."[3] Since the comparisons – the metaphors – in *Hacking Literature* are not perfect, its readers can, as Frost suggests, use the imperfections of its metaphors as opportunities for further exploration concerning why the metaphors break and what those breakages mean.

Hacking Literature's juxtaposed concepts result in repeated references to "digital technologies" and "literary technologies" or "technologies of

[2] Ibid. 3.
[3] Ibid. 4.

literature" as though they are two focused subjects with clearly defined boundaries. But, of course, they are not so simply demarcated. Digital technologies, the younger of the two technologies, encompass thousands of different industries, professions, utilities, and knowledge bases. Lumping those numerous components into two words oversimplifies the substantial underlying complexities. Consider, for example, the complexities incorporated within the scope of the digital topics presented in this project: a software development paradigm undergirding countless programs; an open source operating system kernel powering a majority of the Internet; mechanisms that help maintain security across the Internet; the history of the Internet's most prominent network – the World Wide Web; and the Internet's most successful blogging engine, WordPress, which runs millions of websites. Each of these topics can be (and are) the subject of many books.

The concept of "literary technologies" is even more expansive. If the metaphor of this project's primary comparisons can continue to be extended to questions of scope, then the overwhelming complexity of a concept as young as digital technologies should point to the unimaginable complexities of the topic I describe as "literary technologies." The vastness of information and knowledge contained within a concept like "literary technologies" is visible in the temporal scale of *Hacking Literature*'s literary subjects. The time stretching between the project's first author/text – Homer – and its last author/text – Dickinson – spans some 3,000 years. By contrast, this document barely spans 300 pages and obviously does not have enough space to include all of the complexities to which it alludes or on which it draws.

But even if 3,000 pages or 30,000 pages, *Hacking Literature* would not be, as the title of this final section implies, "concluded." How could the discussion of topics so broad and complex be complete? This impossibility can account for many of the project's textual and conceptual limitations and also offers *Hacking Literature*'s greatest opportunities. Those opportunities exist in the breaking of metaphors.

The breaking of metaphors does not weaken the usefulness of education by metaphor; the breaking of metaphors is the moment when metaphoric technology – a technology presenting similarities – gives way to demonstrating differences. As are the recognition of similarities, those differences are educational. If not – if everything were the same – nothing new would need to be learned.

The value of breaking metaphors is, itself, a metaphoric opportunity that helps explain *Hacking Literature's* purpose and organization. Breaking literary texts "hacks" them through a process of dismantling a system at points of failure in order to see how the system works. While sometimes the process of breaking offers little value, at other times those points of breakage present opportunities for innovation. Examples of these kinds of innovations in response to technological failure appear in each of this project's chapter-level examples of literary technologies. My second chapter, "Developing the Homeric Question," speculates about the reason "original" Homer no longer exists and hypothesizes that is because original Homer – whoever or whatever that was – did not continue to serve the needs of its users, so users updated the technology. "There Is No World Without Verona Firewalls," the third

chapter, speculates whether users of the "lovers from feuding households" framework needed more stability, so uses the author function Shakespeare to help stabilize the technology. The fourth chapter, "This Particular World Wide Web," sees Eliot's difficulty with beginnings and endings as a struggle with the artificial boundaries of print-based technologies, and posits that Eliot attempts to overcome these limitations by linking her text to other texts and building a metatext. The fifth chapter, "Emily Dickinson, Software Engineer," views the nineteenth-century Amherst poet's struggle with the many "inflections" of the pen as a technological failure she attempts to overcome by deploying programmatic language.

Because moments of technological breakage also appear to be the points of innovation, then the goal of a process like hacking literature is to uncover both points where literary technology has already failed, and also where future failures might occur. This process reveals points of previous innovation as well as opportunities for future innovations, and also suggests that the point at which a technology fails is the point at which an art begins. As programmer Emily Dickinson knew, the "World" of this text "is not conclusion" (FP373).

Appendices

Appendix 1: Emily Dickinson Manuscript Links

1. While it is Alive (FP287)
   http://www.edickinson.org/editions/1/image_sets/72754

2. For Each Ecstatic Instant (FP109)
   http://www.edickinson.org/editions/1/image_sets/80195

3. Success is Counted Sweetest (FP112)
   http://www.edickinson.org/editions/1/image_sets/70455

4. I came to buy a smile – today – (FP258)
   http://www.edickinson.org/editions/1/image_sets/69674

5. If I can stop one heart from breaking – (FP982)
   http://www.edickinson.org/editions/1/image_sets/69805

6. If you were coming in the fall (FP356)
   http://www.edickinson.org/editions/1/image_sets/69042

7. How the old mountains drip with sunset (FP327)
   http://www.edickinson.org/editions/1/image_sets/68965

8. You see I cannot see – your lifetime – (FP313)
   http://www.edickinson.org/editions/1/image_sets/68949

9. To die – takes just a little while – (FP315)
   http://www.edickinson.org/editions/1/image_sets/68951

10. She sweeps with many-colored Brooms – (FP318)
    http://www.edickinson.org/editions/1/image_sets/68954

11. There's a certain Slant of light, (FP320)
    http://www.edickinson.org/editions/1/image_sets/68956

12. Blazing in Gold – and – (FP321)
    http://www.edickinson.org/editions/1/image_sets/68957

13. Goof Night! Which put the Candle out? (FP322)
    http://www.edickinson.org/editions/1/image_sets/68958

Bibliography

_____. “About Wordpress.” WordPress.org. Web. 19 Mar. 2014. http://wordpress.org/about/.

_____. “Architecture of the World Wide Web, Volume One.” *W3C*. 15 Dec. 2004. Web. 17 Jul. 2012. http://www.w3.org/TR/webarch/.

Arthur, W. Brian. “Competing Technologies, Increasing Returns, and Lock-In by Historical Events.” *The Economic Journal* , Vol. 99, No. 394 (Mar., 1989), pp. 116-131.

Auerbach, Nina. “Dorothea’s Lost Dog.” Middlemarch *in the 21st Century*. ed. Karen Chase. Oxford: Oxford University Press, 2006.

Bakker, Egbert J. "Homer, Hypertext, and the Web of Myth." Web. 5 Oct. 2011. http://www.philo.umontreal.ca/documents/cahiers/Bakker_homer.pdf.

Beaty, Jerome. Middlemarch *from Notebook to Novel: A Study of George Eliot’s Creative Mind*. Urbana, IL: University of Illinois Press, 1960.

Bedient, Calvin. "*Middlemarch*: Touching Down." *The Hudson Review*, Vol. 22, No. 1 (Spring, 1969): 70-84.

Beer, Gillian. *Darwin’s Plots: Evolutionary Narrative in Darwin, George Eliot and Nineteenth-Century Fiction*. Cambridge: Cambridge University Press, 2009.

Beer, Gillian. “What’s Not in *Middlemarch*.” Middlemarch *in the 21st Century*. ed. Karen Chase. Oxford: Oxford University Press, 2006.

Berners-Lee, Tim. “A Brief History of the Web.” 1993. Web. 14 Jul. 2012. http://www.w3.org/DesignIssues/TimBook-old/History.html.

Berners-Lee, Tim. “Information Management: A Proposal.” May 1990. Web. 13 Jul. 2012. http://www.w3.org/History/1989/proposal.html.

Boone, Joseph Allen. *Tradition Counter Tradition: Love and the Form of Fiction*. Chicago: University of Chicago Press, 1989.

Borges, Jorge Luis. *The Library Of Babel*. Trans. J.E.I. Web. 5 Oct. 2011. http://jubal.westnet.com/hyperdiscordia/library_of_babel.html.

Bowling, Lawrence Edward. “The Thematic Framework of *Romeo and Juliet*.” *PMLA,* Vol. 64, No. 1 (Mar., 1949): 208-220.

Bush, Douglas. *Mythology and the Renaissance Tradition in English Poetry*. 1932; reprint New York, 1957.

Cameron, Sharon. *Choosing Not Choosing*. Chicago, IL: The University of Chicago Press, 1992.

Campbell, Joseph. *The Hero With A Thousand Faces*. New York: MJF Books, 1949.

Chase, Karen, ed. Middlemarch *in the 21st Century*. Oxford: Oxford University Press, 2006.

Clarke, Howard. *Homer's Readers: A Historical Introduction to the* Iliad *and the* Odyssey. Newark, DE: University of Delaware Press, 1980.

Clayton, Jay. *Charles Dickens in Cyberspace*. New York, NY: Oxford University Press, 2003.

Clement of Alexandria*, Exhortation to the Hellenes* 4.52, trans, G. W. Butterworth, Loeb Classical Library (1999).

_____. “Company.” *Google*. Google, Inc. Web. Feb 9, 2013. http://www.google.com/about/company/

Dante. *La Divina Commedia*. ed. D. Lodovico Di Borbone. 1796.

Demetriou, Tania. "'Essentially Circe': Spenser, Homer, and the Homeric Tradition." *Translation and Literature* Vol. 15, No. 2 (Autumn, 2006): 151-176.

Derrida, Jacques. *Archive Fever*. Chicago: The University of Chicago Press, 1996.

Dickinson, Emily. *The Manuscript Books of Emily Dickinson, Volume I*. ed. Ralph Franklin. Cambridge, MA: The Belknap Press of Harvard University Press, 1981.

Dickinson, Emily. *The Manuscript Books of Emily Dickinson, Volume II*. ed. Ralph Franklin. Cambridge, MA: The Belknap Press of Harvard University Press, 1981.

Dimock, Wai Chee. *Through Other Continents: American Literature Across Deep Time*. Princeton, NJ: Princeton University Press, 2008.

Eckstein, Robert. “Java SE Application Design With MVC.” Oracle. Mar. 2007. Web. 10 Jul. 2012. http://www.oracle.com/technetwork/articles/javase/mvc-136693.html.

Eichin, Mark W. and Jon A. Rochlis. “With Microscope and Tweezers: Analysis of the Internet Virus of November 1988.” 9 Feb. 1989. Web. 9 Feb 2013.

Eisenstein, Elizabeth. *The Printing Press As An Agent of Change*. Cambridge, UK: Cambridge University Press, 1997.

Einstein, Albert. *The World As I See It*. Minneapolis: Filiquarian Publishing, 2006.

Eliot, George. *Middlemarch*. New York and Boston: H. M. Caldwell Company Publishers.

Eliot, George. "Notes on Form in Art." fr. *George Eliot: Selected Essays, Poems, and Other Writings*. New York: Penguin Classics, 1991.

Eliot, George. *Quarry For* Middlemarch. ed. Anna Theresa Kitchel. Berkely, CA: University of California Press, 1950.

_____. *Emily Dickinson Archive*. Web. 17 Mar. 2014. http://edickinson.org.

Farrell, Joseph. "Roman Homer." Fr. *The Cambridge Companion to Homer*. Ed. Robert Fowler. Cambridge: Cambridge University press, 2004.

Foley, John Miles. *Homer's Traditional Art*. University Park, PA: Pennsylvania State University Press, 1999.

Foley, John Miles. "'Reading' Homer through Oral tradition." *College Literature*. Vol. 34, No. 2, Reading Homer in the 21st Century (Spring, 2007): 1-28.

Folsom, Ed. "Database as Genre: The Epic Transformation of Archives." *PMLA* 122.5 (2007): 1571-1579

Folsom, Ed. "Reply." *PMLA* 122.5 (2007): 1608-1612.

Ford, Philip. "Homer in the French Renaissance." *Renaissance Quarterly* 59.1 (2006): 1-28.

Foucault, Michel. “What Is an Author?” *Language, Counter-memory, Practice: Selected Essays and Interviews*. ed. Donald F. Bouchard. Ithaca: Cornell University Press, 1980. 113-138.

Fowler, Robert (ed.), *The Cambridge Companion to Homer*. Cambridge: Cambridge University Press, 2004.

Freedman, Florence Bernstein. *Walt Whitman Looks at the Schools*. King’s Crown Press, New York, NY, 1950.

Freedman, Jonathan. “Whitman, Database, Information Culture.” *PMLA* 122.5 (2007): 1596 1603.

Freud, Sigmund. *Beyond the Pleasure Principle*. Ed. James Strachey. New York: W.W. Norton & Company, Inc. 1990.

Frost, Robert. “Education by Poetry.” *Amherst Graduate’s Quarterly*. Feb. 1931. Web. 17 Apr. 2014. http://moodyap.pbworks.com/f/frost.EducationByPoetry.pdf.

Galloway, Alexander R. *Protocl: How Control Exists After Decentralization*. Cambridge, MA: MIT Press, 2004.

Gamma, Erich, Richard Helm, Ralph Johnson, and John Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software*. Boston: Addison-Wesley Professional, 1994.

Gervais, Bertrand. "Is There a Text on This Screen? Reading in an Era of Hypertextuality." *A Companion to Digital Literary Studies*. New York, NY: Wiley-Blackwell, 2008.

Gilbert, Sandra M. “‘If a lion could talk…’: Dickinson Translated.” *The Emily Dickinson Journal*, Vol. 2, No. 2 (Fall, 1993): 1-13.

Ginsburg, Michael Peled. “Pseudonym, Epigraphs, and Narrative Voice: *Middlemarch* and the Problem of Authorship.” *ELH*, Vol. 47, No. 3 (Autumn, 1980): 542-558.

Glaros, Michelle. “<frame>ing Representations of the Web.” *From A to <A>: Keywords of Markup*. ed. Bradley Dilger and Jeff Rice. Minneapolis, MN: University of Minnesota Press, 2010.

Goodheard, Eugene. “The Licensed Trespasser: The Omniscient Narrator in *Middlemarch*.” *The Sewanee Review*, Vol. 107, No. 4 (Fall, 1999): 555-568.

Gurd, Sean Alexander. *Iphigenias At Aulis: Textual Multiplicity, Radical Philology*. Cornell, NY: Cornell University Press, 2005.

Hawker, Philippa. “DiCaprio, DiCaprio, Wherefore art thou, DiCaprio?” *Meanjun*, Vol. 56, No. 1, 1997: 6-16.

Hayles, N. Katherine. *My Mother Was a Computer: Digital Subjects and Literary Texts*. Chicago: The University of Chicago Press, 2005.

Hayles, N. Katherine. “Narrative and Database: Natural Sybionts.” *PMLA* 122.5 (2007): 1603-1608.

Herrera, Brian Eugenio. "Compiling *West Side Story*'s Parahistories, 1949–2009." *Theatre Journal*, 64.2 (2012): 231-247.

Hodgdon, Barbara. “*William Shakespeare’s Romeo + Juliet*: everything’s nice in America?” *Shakespeare and Politics* by Catherine M. S. Alexander. Cambridge: Cambridge University Press, 2004: 247-262.

_____. [Home]. *The Linux Foundation*. Linux Foundation. Web. 16 Jul. 2013. http://www.linuxfoundation.org.

Homer. *The Iliad*. trans. Stephen Mitchell. New York: Atria Books, 2012.

Horace. *The Epistles of Horace*. Ed. Shuckburgh, E. S. Cambridge: Cambridge University Press, 1888.

Howe, Susan. "Experience is the Angled Road." *The Emily Dickinson Journal*, Vol. 15, No. 2, 2006: 34-37.

Howe, Susan. *My Emily Dickinson*. Berkeley, California: North Atlantic Books, 1985.

_____. *Internet Archive*. Internet Archive. Web. 9 Feb. 2013. http://archive.org.

_____. "IBM Watson." *IBM*. IBM, Inc. Web. 9 Feb. 2013. http://www-03.ibm.com/innovation/us/watson/what-is-watson/the-next-grand-challenge.html.

James, Henry. fr. Review in *Galaxy*. ed. William Conant Church. Vol. 15 (Jan. 1873 - Jun. 1873): 424-428.

Jennings, Pamela. "Narrative Structures for New Media: Towards a New Definition." *Leonardo*, Vol. 29, No. 5, Fourth Annual New York Digital Salon (1996): 345-350.

Jones, Brandon. "WordPress Cheat Sheets: The Loop Visual Model." Wptuts+. Nettuts. 18 Oct. 2011. Web. 8 Feb. 2013. http://wp.tutsplus.com/articles/wordpress-cheat-sheets-the-loop-visual-model/.

Kirschenbaum, Matthew. *Mechanisms: New Media and the Forensic Imagination*. Boston: MIT Press, 2008.

Klammer, Martin. *Whitman, Slavery, and the Emergence of* Leaves of Grass. University Park, PA: The Pennsylvania State University Press, 1995.

Lakoff, George and Mark Johnson. *Metaphors We Live By*. Chicago, IL: The University of Chicago Press, 2003.

Lanier, Jaron. *You Are Not A Gadget*. New York, NY: Vintage Books, 2011.

Learmonth, Michael. "Giving It All Away." *Metroactive*. 8 May 1997. Web. 9 Feb. 2013. http://www.metroactive.com/papers/metro/05.08.97/cover/linus-9719.html

Lehmann, Courtney. "Strictly Shakespeare? Dead Letters, Ghostly Fathers, and the Cultural Pathology of Authorship in Baz Luhrmann's 'William Shakespeare's Romeo + Juliet'." *Shakespeare Quarterly*. Vol. 52, No. 2 (Summer, 2001): 189-221.

Leiner, Barry M. et al. "Brief History of the Internet." *Internet Society*. Web. 18 Jul. 2012. http://www.internetsociety.org/internet/internet-51/history-internet/brief-history-internet.

Levenson, Jill L. "Romeo and Juliet before Shakespeare." *Studies in Philology*, Vol. 81, No. 3 (Summer, 1984): 325-347.

Liu, Alan. "Friending the Past: The Sense of History and Social Computing." *New Literary History*. 42.1 (2011): 1-30.

Loewenstein, Joseph. *Ben Jonson and Possessive Authorship*. Cambridge: Cambridge University Press, 2002.

Loewenstein, Joseph. *The Author's Due: Printing and the Prehistory of Copyright*. Chicago: University of Chicago Press, 2002.

_____. “Longer Biography.” Web. 14 Jul. 2012. http://www.w3.org/People/Berners-Lee/Longer.html.

Mackie, Hilary. "The Key to Epic Life?: Classical Study in George Eliot's *Middlemarch*." *Classical World* 103.1 (2009): 53-67.

Manovich, Lev. *The Language of New Media*. Cambridge, MA: MIT Press, 2001.

Mansell, Jr., Darrel. “George Eliot’s Conception of ‘Form’.” *Studies in English Literature, 1500-1900*, Vol. 5, No. 4, Nineteenth Century (Autumn, 1965): 651-662.

Maureen N. McLane. and Laura M. Slatkin. "British Romantic Homer: Oral Tradition, "Primitive Poetry" and the Emergence of Comparative Poetics in Britain, 1760-1830." ELH 78.3 (2011): 687-714.

Mayer-Schonberger, Viktor. *Delete: The Virtue of Forgetting in the Digital Age*. Princeton, NJ: Princeton University Press, 2009.

McGann, Jerome. *Black Riders: The Visible Language of Modernism*. Princeton, NJ: Princeton University Press, 1993.

McGann, Jerome. “Database, Interface, and Archival Fever.” *PMLA* 122.5 (2007): 1598-1592.

McGill, Meredith L. “Remediating Whitman.” *PMLA* 122.5 (2007): 1592-1596.

McLaughlin, Brett D. *Head First Object-Oriented Analysis and Design*. Sebastopol, CA: O’Reilly Media, 2006.

Miller, Cristanne. *Reading In Time: Emily Dickinson in the 19th Century*. Amherst, MA: University of Massachusetts Press, 2012.

Miller, D. A. “D. A. Miller on perspective and Dorothea at the Window.” *George Eliot: Comprehensive Research and Study Guide, Bloom’s Major Novelists*. ed. Harold Bloom. New York: Chelsea House Publishers, 2003.

Miller, J. Hillis. “A Conclusion in Which Almost Nothing is Concluded.” What’s Not in *Middlemarch*.” Middlemarch *in the 21st Century*. ed. Karen Chase. Oxford: Oxford University Press, 2006.

Mitchell, Domhnall. “The Grammar of Ornament: Emily Dickinson’s Manuscripts and Their Meanings.” *Nineteenth-Century Literature*, Vol 55, No. 4 (March 2001): 479-514.

Moore, Olin H. *The Legend of Romeo and Juliet*. Columbus, OH: The Ohio State University Press.

Mueller, Martin. “Digital Shakespeare, or towards a literary informatics.” Shakespeare, Vol.4, No. 3, September 2008. 9-10.

Nardo, Anna K. *George Eliot’s Dialogue with John Milton*. Columbia, MO: University of Missouri Press, 2003.

Nelson, Ted. *Hyperland.com*. Web. 18 Jul. 2012. http://hyperland.com/TBLpage.

Nietzsche. "Homer and Classical Philology" tr. John McFarland Kennedy. Web. 9 Feb. 2013. http://www.gutenberg.org/files/18188/18188-h/18188-h.htm.

Novy, Marianne. “*Middlemarch* and George Eliot’s Female (Re)Vision of Shakespeare.” *The Journal of English and Germanic Philology*, Vol. 90, No. 1 (Jan., 1991): 61-78.

Ostriker, Alicia. “Re-Playing the Bible: My Emily Dickinson.” Fr. *Companion to Emily Dickinson*. Ed. Martha Nell Smith. New York, NY: Wiley-Blackwell,

Parry, Millman. *The Making of Homeric Verse: The Collected Papers of Milman Parry*. Ed. Adam Parry. Oxford: Oxford University Press, 1971.

Passannante, Gerard. "Homer Atomized: Francis Bacon and the Matter of Tradition." *ELH* 76.4 (2009): 1015-1047.

Porter, James I. “Homer: The History of an Idea.” Fr. *The Cambridge Companion to Homer*. Ed. Robert Fowler. Cambridge: Cambridge University press, 2004.

Porter, James I. "Homer: The Very Idea." *Arion: A Journal of Humanities and the Classices*, Third Series, Vol. 10, No. 2 (Fall, 2002), 57-86.

Powers, William. *Hamlet's Blackberry: A Practical Philosophy for Building a Good Life in the Digital Age*. New York: Harper, 2010.

Pratt, John Clark and Victor A. Neufeldt. *George Eliot’s* Middlemarch *Notebooks: A Transcription*. Berkeley, CA: University of California Press, 1979.

Rader, Ralph W. “The Comparative Anatomy of Three Baggy Monsters: *Bleak House*, *Vanity Fair*, *Middlemarch*.” *The Journal of Narrative Technique*, Vol. 19, No. 1 (Winter, 1989): 46-69.

Ramsay, Stephen. *Reading Machines: Toward an Algorithmic Criticism*. Urbana, IL: University of Illinois Press, 2011.

Reenskaug, Trygve. “Models - Views - Controllers.” 10 Dec. 1979. PDF File. http://heim.ifi.uio.no/~trygver/1979/mvc-2/1979-12-MVC.pdf.

Reenskaug, Trygve. “MVC XEROX PARC 1978-79.” n.p. n.d. Web. 8 Feb. 2013. http://heim.ifi.uio.no/~trygver/themes/mvc/mvc-index.html.

Reenskaug, Trygve. “Thing-Model-View-Editor: and Example from a planningsystem.” 12 May 1979. PDF File. http://heim.ifi.uio.no/~trygver/1979/mvc-1/1979-05-MVC.pdf

Reynolds, J. “The Helminthiasis of the Internet.” Network Working Group RFC. Dec. 1989. Web. 9 Feb. 2013. http://tools.ietf.org/html/rfc1135.

Rich, Adrienne. “Vesuvius at Home: The Power of Emily Dickinson (1975).” *On Lies, Secrets, and Silence*. New York: W.W. Norton & Company, 1979.

Rickert, Thomas. “Tarrying With The <head>: The Emergence of Control Through Protocol.” *From A to <A>: Keywords of Markup*. ed. Bradley Dilger and Jeff Rice. Minneapolis, MN: University of Minnesota Press, 2010.

Rothstein, Marian. "Homer for the Court of François I" *Renaissance Quarterly*. 59.3 (2006): 732-767.

Rukeyser, Muriel. *The Life of Poetry*. Ashfield, MA: Paris Press, 1996.

Russell, David R. and Arturo Yañez. “‘Big Picture People Rarely Become Historians’: Genre Systems and the Contradictions of General Education.” Fr. *Writing Selves/Writing Society*. Ed. Charles Bazerman and David R. Russell. Fort Collins, CO: WAC Clearinghouse, 2003.

Salel, Hugues. "Dedicatory Epistle to Homer's *Iliad*," II. 145-50. Translation fr. Bernard Weinberg. *Critical Prefaces of the French Renaissance*. 1950. Reprinted New York, 1970.

Sciarrino, Enrica. "The Introduction of Epic in Rome: Cultural Thefts and Social Contests." *Arethusa* 39.3 (2006): 449-469.

Sedgwick, Eve Kosofsky. *Touching Feeling: Affect, Pedagogy, Performativity*. Durham, NC: Duke University Press, 2003.

_____. *SEOMoz*. SEOmoz, Inc. Web. 9 Feb. 2013. http://www.seomoz.org.

Shakespeare, William. *Romeo and Juliet*. ed. Joseph Pearce. San Francisco: Ignatius Press, 2011.

Sircy, Otice C. "'The Fashion of Sentiment': Allusive Technique and the Sonnets of *Middlemarch*." *Studies in Philology*, Vol. 84, No. 2 (Spring, 1987): 219-244.

Smith, Martha Nell and Lara Vetter. “Emily Dickinson Writing a Poem.” Web. 17 Mar. 2014. http://archive.emilydickinson.org/safe/index.html

Smith, Martha Nell (ed). *Dickinson Electronic Archives*. 1994-2012. Web. 10 Feb. 2013. http://emilydickinson.org.

Smith, Martha Nell. “The Human Touch – Software of the Highest Order: Revisiting Editing as Interpretation,” presented at Digital Textual Studies: Past, Present and Future, Texas A&M University Department of English, October 2006.

Smith, Martha Nell and Ellen Louise Hart. *Open Me Carefully: Emily Dickinson’s Intimate Letters to Susan Huntington Dickinson*. Ashfield, MA: Paris Press, 1998.

Smith, Martha Nell and Lara Vetter. *Emily Dickinson's Correspondences: A Born-Digital Textual Inquiry*. Charlottesville, VA: The University of

Virginia Press, 2008. Web. 5 Oct. 2011. http://rotunda.upress.virginia.edu/edc/.

Socarides, Alexandra. *Dickinson Unbound: Paper, Process, Poetics*. Oxford: Oxford University Press, 2012.

Sowerby, Robin. "Early Humanist Failure with Homer (I)." *International Jounral of the Classical Tradition*. Vol. 4, No. 1 (Summer, 1997): 37-63.

Sowerby, Robin. "Early Humanist Failure with Homer (II)." *International Jounral of the Classical Tradition*. Vol. 4, No. 2 (Fall, 1997): 165-194.

Spafford, Eugene H. "The Internet Worm Program: An Analysis." Technical Report. 8 Dec. 1988. Web. 9 Feb. 2013. http://spaf.cerias.purdue.edu/tech-reps/823.pdf.

_____. *Such Tweet Sorrow*. The Royal Shakespeare Company, 2010. Web. 5 Oct. 2011. http://suchtweetsorrow.com.

_____. "Summing Up 3.2." *The H*. Heise Media UK Ltd. n.d. Web. 9 Feb. 2013. http://www.h-online.com/open/features/What-s-new-in-Linux-3-2-1400680.html?page=3.

Stallybrass, Peter. "Against Thinking." *PMLA* 122.5 (2007): 1580-1587.

Swan, Brian. "*Middlemarch* and Myth." *Nineteenth-Century Fiction*, Vol. 28, No. 2 (Sep., 1973): 210 – 214.

Teskey, Gordon. "Homer," in *The Spenser Encyclopedia*, edited by A. G. Hamilton. Toronto, 1990.

Thomson, James B. *Practical Arithmetic*. New York: Ivison, Phinney & Company, 1847.

Torvalds, Linus. “Free minix-like kernel sources for 386-AT.” Fr. comp.os.minix. 5 Oct. 1991. Web. 9 Feb. 2013. https://groups.google.com/forum/?fromgroups=#!topic/comp.os.minix/4995SivOl9o.

Turkle, Sherry. *Alone Together: Why We Expect More from Technology and Less from Each Other*. New York, NY: Basic Books, 2011.

_____. “URIs, URLs, and URNs: Calrifications and Recommendations 1.0.” The World Wide Web Consortium (W3C). Web. 13 Aug. 2013. http://www.w3.org/TR/uri-clarification/.

Vallianatos, E. G. "Christians and the Classics: War against Reason." *Mediterranean Quarterly* 15.3 (2004): 75-94.

Van Der Horst, Pieter W. "Plato's Fear as a Topic in Early Christian Apologetics." *Journal of Early Christian Studies* 6.1 (1998): 1-14.

Werner, Marta. *Emily Dickinson’s Open Folios: Scenes of Reading, Surfaces of Writing*. Ann Arbor, MI: University of Michigan Press, 1996.

Werner, Marta. “Ravished Slates.” Web. 17 Mar. 2014. http://www.emilydickinson.org/ravished-slates-re-visioning-the-lord-letters

West, David. *Object Thinking*. Seattle: Microsoft Press, 2004.

Whitman, Walt. *Leaves of Grass and Other Writings*. Ed. Michael Moon. New York: W.W. Norton & Company, Inc., 2002.

Williams, R. D. "Virgil and the 'Odyssey'." *Phoenix*. Vol. 17, No. 4 (Winter, 1963): 266-274.

Wilson, Penelope. "Homer and English Epic" fr. *The Cambridge Companion to Homer*. Ed. Robert Fowler. Cambridge: Cambridge University Press, 2004.

Wolf, F. A. *Prolegomena To Homer*. Trans. Anthony Grafton, Glenn W. Most, and James E. G. Zetzel. Princeton, NJ: Princton University Press, 1985.

Select Works Consulted

Digital Media

Ascott, Roy. "Technoetic Pathways Toward the Spiritual in Art: A Transdisciplinary Perspective on Connectedness, Coherence, and Consciousness." *Leonardo*. Vol. 39, No. 1 (February, 2006): 65-69.

Bartlett, Laura, and Thomas B. Byers. "Back to the Future: The Humanist 'Matrix.'" *Cultural Critique*. No. 53, Posthumanism (Winter, 2003): 28-46.

Bay, Jennifer L. "Body On <body>." *From A to <A>: Keywords of Markup*. ed. Bradley Dilger and Jeff Rice. Minneapolis, MN: University of Minnesota Press, 2010.

Burgess, Helen J. "<?php>: 'Invisible' Code and the Mystique of Web Writing." *From A to <A>: Keywords of Markup*. ed. Bradley Dilger and Jeff Rice. Minneapolis, MN: University of Minnesota Press, 2010.

Clayton, Jay. "Convergence of the Two Cultures: A Geek's Guide to Contemporary Literature." *American Literature*. Vol. 74, No. 4 (December, 2002): 807-831.

Davidson, Cathy. *Now Your See It: How the Brain Science of Attention Will Transform the Way We Live, Work, and Learn*. New York, NY: Viking, 2011.

Drucker, Johanna. "Speculative Computing: Aesthetic Provocations in Humanities Computing." *A Companion to Digital Humanities*. New York, NY: Wiley-Blackwell, 2005.

Cummings, James. "The Text Encoding Initiative and the Study of Literature." *A Companion to Digital Literary Studies*. New York, NY: Wiley-Blackwell, 2008.

Ess, Charles. "'Revolution? What Revolution?' Successes and Limits of Computing technologies in Philosophy and Religion." *A Companion to Digital Humanities*. New York, NY: Wiley-Blackwell, 2005.

Facklet, Maria Francesca. "'I'll Google It': Gossip, Queer Intimacies, and the Internet." *Modern Drama*, Vol. 53, No. 3. University of Toronto Press, 2010. p. 390-409.

Forster, E.M. "The Machine Stops." Halcyon Classics, 2010.

Haraway, Donna. "A Cyborg Manifesto: Science, Technology, and Socialist-Feminism in the Late Twentieth Century," in *Simians, Cyborgs and Women: The Reinvention of Nature* (New York; Routledge, 1991), pp.149-181.

Hockey, Susan. "The History of Humanities Computing." *A Companion to Digital Humanities*. New York, NY: Wiley-Blackwell, 2005.

Hodgdon, Barbara. "(You)Tube Travel: The 9:59 to Dover Beach, Stopping at Fair Verona and Elsinore." *Shakespeare Bulletin*, Vol. 28, No. 3. Johns Hopkins University Press, 2010. p. 313-330.

Benjamin, Walter. *Illuminations*. Harry Zohn, Trans. New York, NY: Schoken Books, 1968.

Jensen, Michael. "Intermediation and its Maclontents: Validating Professionalism in the Age of Raw Dissemination." *A Companion to Digital Humanities*. New York, NY: Wiley-Blackwell, 2005.

Kirschenbaum, Matthew G. "'So the Colors Cover the Wires': Interface, Aesthetics, and Usability." *A Companion to Digital Humanities*. New York, NY: Wiley-Blackwell, 2005.

Manovich, Lev. "Media After Software." *Journal of Visual Culture*. Vol. 12, No. 1 (April, 2013).

McGann, Jerome. "Marking Texts of Many Dimensions." *A Companion to Digital Humanities*. New York, NY: Wiley-Blackwell, 2005.

Moretti, Franco. *Graphs, Maps, Tress: Abstract Models for Literary History*. Verso: 2007.

Price, Kenneth M. "Electronic Scholarly Editions." *A Companion to Digital Literary Studies*. New York, NY: Wiley-Blackwell, 2008.

Ramsay, Stephen. "Algorithmic Criticism." *A Companion to Digital Literary Studies*. New York, NY: Wiley-Blackwell, 2008.

Reilly, Colleen A. "alt: Accessible Web Design or Token Gesture?" *From A to <A>: Keywords of Markup*. ed. Bradley Dilger and Jeff Rice. Minneapolis, MN: University of Minnesota Press, 2010.

Rice, Jeff. “English <A>.” *From A to <A>: Keywords of Markup*. ed. Bradley Dilger and Jeff Rice. Minneapolis, MN: University of Minnesota Press, 2010.

Riley, Brendan. “A Style Guide to the Secrets of <style>.” *From A to <A>: Keywords of Markup*. ed. Bradley Dilger and Jeff Rice. Minneapolis, MN: University of Minnesota Press, 2010.

Ryan, Marie-Laure. "Multivariant Narratives." *A Companion to Digital Humanities*. New York, NY: Wiley-Blackwell, 2005.

Shirky, Clay. *Here Comes Everybody: The Power of Organizing Without Organizations*. New York, NY: Penguin Books, 2009.

Smith, Martha Nell. "Electronic Scholarly Editing." *A Companion to Digital Humanities*. New York, NY: Wiley-Blackwell, 2005.

Tietz, Ward. “Linking and Care in Connection.” *New Literary History*. Vol. 35, No. 3, Critical Inquiries, Explorations, and Explanations (Summer, 2004): 507-522.

Author-Specific Texts

Aicher, Peter. “Ennius’ Dream of Homer.” *The American Journal of Philology*. Vol. 110, No. 2 (Summer, 1989): 227-232.

Bonnard, Georges A. “*Romeo and Juliet*: A Possible Significance.” *The Review of English Studies*. Vol. II, No. 5 (1951): 319-327.

Bloom, Harold. *Shakespeare: The Invention of the Human*. New York, NY: Riverhead Books, 1998.

Bloom, Harold, *Ed. William Shakespeare's* Romeo and Juliet. Philadelphia, PA: Chelsea House Publishers: 2000.

Combellack, Frederick M. "Homer the Innovator." *Classical Philology*. Vol. 71, No. 1 (Jan., 1976): 44-55.

Doriani, Beth Maclay. "Emily Dickinson, Homiletics, and Prophetic Power." *The Emily Dickinson Journal*. Vol. 1, No. 2 (1992): 54-75.

Emerson, Lori. "My Digital Dickinson." *The Emily Dickinson Journal*. Vol. 17, No. 2 (2008): 55-76.

Finch, A. R. C. "Dickinson and Patriarchal Meter: A Theory of Metrical Codes." *PMLA*. Vol. 103, No. 2 (March, 1987): 166-176.

Gatens, Moira. "The Art and Philosophy of George Eliot." *Philosophy and Literature*. Vol. 33, No. 1 (2009): 73-90.

Gerlach, John. "Reading Dickinson: Bolts, Hounds, the Variorum, and Fascicle 39." *The Emily Dickinson Journal*. Vol. 3, No. 2 (1994): 78-99.

Gindele, Karen C. "The Web of Necessity: George Eliot's Theory of Ideology." *Texas Studies in Literature and Language*. Vol. 42, No. 3 (Fall, 2000): 255-289.

Hernandez, Pura Nieto. "Reading Homer in the 21st Century." *College Literature*. Vol. 34, No. 2 (Spring, 2007): 29-54.

Heginbotham, Eleanor Elson. *Reading the Fascicles of Emily Dickinson: Dwelling in Possibilities*. Columbus, OH: Ohio State University Press, 2003.

Hulme, Hilda M. "*Middlemarch* as Science-Fiction: Notes on Language and Imagery." *Novel: A Forum on Fiction*. Vol. 2, No. 1 (Autumn, 1968): 36-45.

Hughes, Linda K. "Constructing Fictions of Authorship in George Eliot's *Middlemarch*, 1871-1872." *Victorian Periodicals Review*. Vol. 38, No. 2 (Summer, 2005): 158-179.

Kottman, Paul A. "Defying the Stars: Tragic Love as the Struggle for Freedom in *Romeo and Juliet*." *Shakespeare Quarterly*. Vol. 63, No. 1 (2012): 1-38.

Kawashima, Robert S. "Verbal Medium and Narrative Art in homer and the Bible." *Philosophy and Literature*. Vol. 28, No. 1 (April, 2004): 103-117.

Law, Robert Adger. "On Shakespeare's Changes of His Source Material in *Romeo and Juliet*." *Studies in English*. No. 9 (July, 1929): 86-102.

Levenson, Jill L. "The Definition of Love: Shakespeare's Phrasing in *Romeo and Juliet*." *Shakespeare Studies*. Vol. 15 (January, 1982): 21-38.

Lucking, David. "And All Things Change Them to the Contrary: *Romeo and Juliet* and the Metaphysics of Language." *English Studies*. Vol. 78, No. 1 (January, 1997): 8-18.

Lucking, David. "Uncomfortable Time in *Romeo and Juliet*." *English Studies*. Vol. 82, No. 2 (2001): 115-126.

MacKenzie, Clayton G. "Love, Sex, and Death in *Romeo and Juliet*." *English Studies*. Vol. 88, No1. (2007): 22-42.

Maxwell, Catherine. "The Brooking of Desire: Dorothea and Deferment in *Middlemarch*." *The Yearbook of English Studies*. Vol. 26 (1996): 116-126.

Moscovici, Claudia. "Allusive Mischaracterization in *Middlemarch*." *Nineteenth-Century Literature*. Vol. 49, No. 4 (1995): 513-531.

Muir, Kenneth. *The Sources of Shakespeare's Plays*. New Haven: Yale University Press, 1978.

Nevo, Ruth. "Tragic Form in *Romeo and Juliet*." *Studies in English Literature, 1500-1900*. Vol. 9, No. 2, Elizabethan and Jacobean Drama (Spring, 1969): 241-258.

Siegel, Paul N. "Christianity and the Religion of Love in *Romeo and Juliet*." *Shakespeare Quarterly*. Vol. 12, No. 4 (Autumn, 1961): 371-392.

Slater, Candace. "Romeo and Juliet in the Brazilian Backlands." *Journal of Folklore Research*. Vol. 20, No. 1 (May, 1983): 35-53.

Thomson, Leslie. "'With Patient Ears Attend': *Romeo and Juliet* on the Elizabethan Stage." *Studies in Philology*. Vol. 92, No 2. (Spring, 1995): 230-247.

Tucker, John L. "George Eliot's Reflexive Text: Three Tonalities in the Narrative Voice of *Middlemarch*." *Studies in English Literature, 1500-1900*. Vol. 31, No. 4 (Autumn, 1991): 773-791.

Vetter, Lara. "Editing Dickinson in an Electronic Environment." Fr. *A Companion to Emily Dickinson*. Martha Nell Smith and Mary Loeffelholz. Malden, MA: Blackwell Publishing, 2008: 437-452.

Wallace, Nathaniel. “Cultural Tropology in *Romeo and Juliet*.” *Studies in Philology*. Vol. 88, No. 3 (Summer, 1991): 329-344.

Weinberger, Jerry. “Pious Princes and Red-Hot Lovers: The Politics of Shakespeare’s ‘Romeo and Juliet’.” *The Journal of Politics*. Vol. 65: 350-375.

Werner, Marta. “‘For Flash and Click and Suddenness—’: Emily Dickinson and the Photography-Effect.” Fr. *A Companion to Emily Dickinson*. Martha Nell Smith and Mary Loeffelholz. Malden, MA: Blackwell Publishing, 2008: 437-452.

Wilder, Lina Perkins. “Toward a Shakespearean ‘Memory Theater’: Romeo, the Apothecary, and the Performance of Memory.” *Shakespeare Quarterly*. Vol. 56, No. 2 (Summer, 2005): 156-175.

Wylder, Edith. “Emily Dickinson’s Punctuation: The Controversy Revisited.” *American Literary Realism*. Vol. 36, Part 3 (2004): 206-224.

A Note About Wikipedia

While perhaps not as poignant today as a few years ago, concerns regarding the reliability of Wikipedia as a source make me want to give special note of its use in this document. I would like to inclusively cite the thousands of faceless Wikipedia contributors. As a self-taught software developer, while many programming concepts are familiar to me and I use them daily, the exact explanations of concepts like object-oriented and procedural

programming were things I needed to better understand, and Wikipedia was often my first resource for learning more. Although I do not specifically cite any work from Wikipedia, it was often a valuable starting point for learning more about certain topics (especially digital ones), and it even assisted in pointing me toward some of the books and articles ultimately referenced. I would be doing the project and its contributors a great disservice by not noting its immense value here.