HYPERTEXTUAL ULTRASTRUCTURES: MOVEMENT AND CONTAINMENT IN TEXTS AND HYPERTEXTS

A Dissertation

by

ROSEMARIE L. COSTE

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

August 2009

Major Subject: English

HYPERTEXTUAL ULTRASTRUCTURES: MOVEMENT AND CONTAINMENT IN TEXTS AND HYPERTEXTS

A Dissertation

by

ROSEMARIE L. COSTE

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Approved by:

Chair of Committee,	M. Jimmie Killingsworth
Committee Members,	Maura C. Ives
	Jacqueline S. Palmer
	Robert R. Shandley
Head of Department,	M. Jimmie Killingsworth

August 2009

Major Subject: English

ABSTRACT

Hypertextual Ultrastructures: Movement and Containment in Texts and Hypertexts.

(August 2009)

Rosemarie L. Coste, B.S., Trinity University;

M.A., The University of Texas at San Antonio

Chair of Advisory Committee: Dr. M. Jimmie Killingsworth

The surface-level experience of hypertextuality as formless and unbounded, blurring boundaries among texts and between readers and writers, is created by a deep structure which is not normally presented to readers and which, like the ultrastructure of living cells, defines and controls texts' nature and functions. Most readers, restricted to surface-level interaction with texts, have little access to the deep structure of any hypertext. In this dissertation, I argue that digital hypertexts differ essentially from paper texts in that hypertexts are constructed in multiple layers, with surface-level appearance and behavior controlled by sub-surface ultrastructure, and that these multiple layers of structure enable and necessitate new methods of textual study designed for digital texts.

Using participant-observation from within my own practice as a webmaster, I closely examine the sub-surface structural layers that create several kinds of Web-based digital hypertexts: blogs, forums, static Web pages, and dynamic Web pages. With these hypertexts as the primary models, along with their enabling software and additional digital texts—wikis, news aggregators, word processing documents, digital photographs, electronic mail, electronic forms—available to me as a reader/author rather than a webmaster, I demonstrate methods of investigating and describing the development of digital texts. These methods, like methods already established within textual studies to trace the development of printed texts, can answer questions about accidental and intentional textual change, the roles of collaborators, and the ways texts are shaped by production processes and mediating technologies. As a step toward a formalist criticism of hypertext, I propose concrete ways of categorizing, describing, and comparing hypertexts and their components. I also demonstrate techniques for visualizing the structures, histories, and interrelationships of hypertexts and explore methods of using self-descriptive surface elements in paper-like texts as partial substitutes for the sub-surface self-description available in software-like texts. By identifying digitization as a gateway to cooperation between human and artificial intelligences rather than an end in itself, I suggest natural areas of expansion for the humanities computing collaboration as well as new methodologies by which originally-printed texts can be studied in their digital forms alongside originally-digital texts.

DEDICATION

This is for Elishalom Yechiel, my husband and friend, who made it possible.

ACKNOWLEDGEMENTS

This project developed in bursts, starting and stopping, then re-starting in a new direction, over many years; the people who, whether they know it or not, helped me to create it are as varied and multi-layered as my hypertextual subject matter.

I'll begin by thanking the large numbers of people I relied on who have no idea how much they helped me: the development, support, and documentation teams for the software I used and examined in this dissertation. The products themselves, and the Web sites from which they can be obtained, are identified throughout the dissertation; of course, all trademarks are the property of their respective owners. For many of the examples and case studies in this dissertation I used, and sometimes dissected, a variety of open-source software tools; of all the excellent reasons to create open-source software, its amenability to in-depth examination by graduate students who want to see and poke at the living guts of a hypertext may be the least appreciated but, to me today, the most important.

My employers, clients, and colleagues have been amazingly patient and flexible as I pursued my education in English while continuing to work in Information Technology. I know that several of the ideas that are central to this dissertation—multi-generational collaboration, active development of alternatives, detailed formal description, sub-surface exploration, and especially the central role of digital texts' human caretakers—come to me directly through my experiences in IT as a programmer, systems programmer, technology planner, webmaster, and technical writer; I could not think the way I think about how humans use language to communicate with each other if

I had not spent so many years learning about how humans use language to communicate with computers.

On the academic side of the equation, my advisory committee shifted shapes over the years as people moved on and the project changed focus. Of the four committee members at my dissertation defense in 2009, only one had also participated in my preliminary exams in 2003: Bob Shandley. I thank him not only for his patience, but for his way of getting me to see that I could pursue, rather than simply mention, important ideas. I know my work on this project is very much clearer and deeper for his support.

Mary Bucholtz, now at the University of California, Santa Barbara, was one of the initial members of my committee; her suggestions for my reading list, especially on Critical Discourse Analysis, strongly influenced my thinking about the value of carefully examining mundane texts such as newspaper stories and government publications. C. Jan Swearingen sat in briefly, replacing Mary in time for my preliminary exams; although I still can't claim to know much about classical rhetoric, her presence inspired me to learn a little about it, and to look for ways to do something similar—describing and naming communicative patterns—in the texts I study.

I am grateful to Maura Ives for introducing me to textual studies in her bibliography class in 1998 and, after joining my committee in 2008, for catching me up on current thinking in textual studies and digital humanities with her detailed suggestions about where else to look and what else to read. Jackie Palmer joined the committee as I was in the final stages of writing my dissertation. The finished product is greatly improved by her close, careful reading, and her insistence that I clarify confusing

points and maintain stronger awareness of specific audiences. Both Jackie and Maura imagined readers and applications for these ideas that, until they pointed them out, I did not; I especially thank them both for that.

There have been three chairs of my advisory committee, each of whom shaped this dissertation in important ways. Finnie Coleman, now at University of New Mexico, read several early proposals and encouraged my interest in the ways stories can be constructed in layers, sometimes by adding new facts on top of old facts, sometimes by wrapping an imaginary world around a central nugget of truth. After Finnie left, Douglas Brooks made up his mind that I was going to graduate and no obstacle between here and there was insurmountable; in his customary stubborn way, he set about pushing people and paperwork into line to make that happen. He wanted to talk about the future—people I should meet, projects I might work on—but saved the specifics for “later,” so as not to distract me from “dissertating.” He died of lung cancer January 27, 2009. I’m sorry he didn’t get to see the finished dissertation, and that I didn’t get to hear the ideas he was saving; I treasure the many he did share. One of my favorite memories of this project is his reaction of amazed laughter when he saw an early attempt at what became Figure 17, relating publication of a digital blog to publication of a Shakespearean play; his “that’s hilarious, that’s beautiful, I love it!” may be the best praise my work has ever received.

Jimmie Killingsworth shared Douglas’ determination that I should, finally, finish my degree. I thank him for many things, beginning with his patient help in trimming a giant pile of ideas down to the size and shape of a dissertation. He did everything he could to make the final months of this project productive, pain-free, and often entertaining. After working with Jimmie, I'm a better writer and thinker; I'm privileged to be his student.

Above and beyond everyone else who helped in large and small ways, I thank my husband, Elishalom Yechiel. When we met, I had just finished my first semester back in school, taking undergraduate English courses "for fun." Soon after we married, I took my preliminary exams and began working on a dissertation; graduation will take place just after our sixth wedding anniversary. I've been in school as long as we've known each other; he's always made sure I know he's proud of me, curious about my ideas, and happy to see me happy. There have been many times when it made no sense for me to continue working on this project, but Eli made sure I kept going anyway. He took on many extra burdens to give me the gifts of time and space to think, read, and write. His energy has made our home a place of intellectual excitement, stimulating conversation, and the joy of investigation and discovery; his insightful questions have helped me learn to organize my thoughts and express my opinions. Eli is the most creative, imaginative, and self-motivated person I know; he is my inspiration and my source of courage. I could not have done this without his love and support.

TABLE OF CONTENTS

CHAPTER Page

LIST OF FIGURES

Page

Page

LIST OF TABLES

CHAPTER I

INTRODUCTION

On the World Wide Web, works of imagination are readily available: fiction, poetry, drama, and film, from the canonical to the experimental, are offered through libraries, archival projects, social networking sites, and creative individuals. Games are also available on the Web, offering both private and communal entertainment experiences. However, works of imagination on the Web are vastly outnumbered by works of information: news agencies provide coverage of current events; governments and organizations announce policies, explain procedures, and process “paperwork”; commercial enterprises describe, sell, and support their products and services. Technical experts in all fields use the Web for its original purpose of supporting research, discussion and collaboration; individuals of varied backgrounds, experts on their own lives if on no other subjects, use the Web to record their experiences and publish their opinions.

To study the Web should be, in some proportion to the pervasiveness of this kind of content, to study the news reports, encyclopedias, policy statements, product catalogs, data entry forms, and other informational texts with which it is populated. To understand these informational digital texts should mean, as it does for the literary and historical texts that have been the focus of textual scholarship, to closely investigate how they are constructed and transmitted, how they can be intentionally or accidentally altered in the

---

This dissertation follows the style of *The MLA Handbook.*

processes of publication, how multiple versions and multiple authorial voices relate to each other, and how social and technological changes alter their structures and uses.

Some of the characteristics of Web-based publication—rapid change, large numbers of versions and collaborators, multiple layers of mediating technology—may have created the assumption that digital texts are too unstable and complex to be usefully examined with anything resembling the rigor with which textual scholars study printed texts; I argue that this is not the case, and that in fact digital texts, often carrying within themselves detailed descriptions of their own histories, can readily provide richly detailed information about their construction, evolution, and use. Interpreting that information and putting it to use in answering scholarly questions about the developmental processes of all texts, including digital texts, is a natural extension of the print-oriented work of textual studies.

Why the Most-Used Parts of the Web Are Studied Least

Scholarship on printing and print culture began by studying the works that made the printing press a commercial success: the Bible and other highly-valued, widely-read (and therefore readily-marketable) works of inspiration and imagination. Many texts and editions and copies of these works were produced, rapidly accumulating into a vast body of material ready-made for a field of study devoted to understanding how they related to each other. Print itself has long been an essential tool not only in the service of literary expression but of practical matters in commerce, the sciences, government, and all other areas of human endeavor; textual studies, however, has devoted little energy beyond the canonical literature with which the field began. In *From Gutenberg to Google*, Peter L.

Shillingsburg, visiting the Wolff collection of 7,000 Victorian novels, mourns the gaps in this and other collections caused by longstanding and pervasive disinterest in informational texts:

> Wolff's collection of books stands in stark isolation from the books that they jostled in the days when they were fresh off the press. Where were the books of agriculture, history, geography, science, religion, politics, cookery, and domestic economy? How could I come to know the meaning or importance or significances of any one of these books without knowing what other books were published by its publishers or what other books were reviewed together with it? (146-147)

Shillingsburg's observation here, that novels and novel readers exist in a world also populated by many kinds of workaday texts which have not been valued and collected and studied with the intensity that has been applied to literary texts, suggests that a more complete context would deepen his understanding of the novels that are his central project. This may be true, although it may also be an unreachable goal: if the less-honored texts that originally accompanied the great works of art from the past have not been preserved, then it will never be possible to re-combine them all into something resembling the textual experience of their first readers and writers. For texts published now, especially those published digitally on the Web, preservation of a rich context becomes not only possible but natural: to extract a Web page from its environment, preserving the highly-valued piece intact but isolated from its connections to surrounding texts and textual components, is a difficult and error-fraught undertaking.

Digital publication began in a direction very different from print publication, with material closer in spirit to the cookbook than to the novel. Electronic word processing began as a cost-saving business tool, more efficient than the typewriter and the file cabinet as a means of creating and storing business letters and memoranda. Databases, too, have their roots in the mundane practicalities of managing product inventories and customer records. The Web was created to simplify and accelerate the exchange of practical, technical information among teams of engineers and scientists, supporting collaboration among geographically-dispersed team members. Artists eventually found ways to adapt all these tools to serve their creative purposes, but the roots of the Web and much of its current usage remain firmly anchored in the useful and instructive more than the decorative and the inspirational. It should be natural, then, that the scholarly examination of hypertexts develop in the same relation to early hypertexts as the study of print did to early printed texts; we should begin by studying the processes of constructing and publishing and maintaining informational hypertexts, with all the care and attention to detail that textual scholars have applied to analyzing early print editions of the Bible and Shakespeare’s plays.

Instead, hypertextuality has been studied within the humanities as an extension of the print-based interests of scholars and researchers. For linguists and rhetoricians, this has meant cataloging new observations of communication styles, including new hybrids of orality and literacy. For literary scholars, studying hypertext literature has meant considering some expanded definitions of storytelling and the relations between authors and readers. For textual scholars, encountering hypertext has meant acquiring access to an array of improved tools with which to collect and examine and compare texts, but it has not meant choosing to examine different texts, and has certainly not repositioned as key objects of study the informational hypertexts that comprise the bulk of the Web itself.

Layers and Ultrastructure: Hypertextuality, Inside-Out

Like the choices made in producing a printed book, the choices implemented in the creation of a hypertext shape its essential nature, including the errors and losses to which it is prone. To demonstrate and describe this, I borrow the concept of *ultrastructure* from cellular biology and demonstrate the validity of its extension into the seemingly-unrelated field of textual studies. "Structure" seems an intuitive concept: complex systems are built, constructed, of simpler components. However, "ultra-"may require some explication.

Applied to the spectrum of light, the prefixes infra- and ultra- define wavelengths imperceptible to unaided human vision. The visible spectrum, the colors we can see within light, is a small portion of light's actual makeup, with red and violet colors representing opposite extremes. Red light emission is at a large wavelength, about 750 nanometers, the largest our eyes can perceive; infrared, which we cannot see but can measure as heat, is larger still. The wavelength of violet light, down to about 380 nanometers, is the smallest our eyes can see; ultraviolet, also invisible to us but evident in the sunburn damage we experience, is even smaller.

Infrastructure is the larger environmental context which supports the activities of individuals. In a city, the infrastructure provides shared resources such as streets, power lines, and emergency rescue services; the infrastructure is owned and maintained by the members of the community, and to a large extent makes possible their existence as a community. Use of the Web is also dependent on access to its infrastructure of networked servers and routers; these resources are publicly accessible but, unlike city streets, are privately owned and maintained. Just as infrared is larger than red, infrastructure is larger than structure; from our perspective within the system, we cannot see it as a whole because it is so much bigger than we are. Ultraviolet is smaller than violet and ultrastructure is smaller than structure; unaided by tools, we can observe ultrastructure's effects but cannot see ultrastructure for the inverse reason, because it is so much smaller than we are.

From any starting point (a Web page, for most of the examples I will discuss), ultrastructure looks inward while the more familiar term, infrastructure, looks outward. While infrastructure is external and communal, ultrastructure is internal and solitary. In cellular biology, electron microscopes make it possible to identify and study cellular ultrastructures that define the cell's nature and enable its activities. For example, cells have organelles which provide them with energy (mitochondria) and mobility (cilia) and which separate them from their environment (membranes). Just as optical and electronic instrumentation makes it possible for biologists to observe intracellular structures and activities that are too small, too rapid, or too complex to be perceived without instrumental support, I propose that visualization tools can provide textual scholars with access to the inner workings of many kinds of digital documents, especially hypertexts. Some of the visualization techniques I propose are already well-established in the field of information technology, as summarized in Stephan Diehl's *Software Visualization: Visualizing the Structure, Behaviour, and Evolution of Software*. Diehl's project demonstrates that visualization tools, created by programmers to make complex scientific and business data—climate and population changes, disease trends, financial transactions—more accessible to non-IT users, can also be used to improve programmers' understanding of their own complex systems. My project in this dissertation is related to Diehl's: by adapting some existing visualization methods and creating some new ones, I think we can improve the quality of description and analysis applied to digital texts, and find ways to deal with the rapid speeds of creation and alteration and the intense levels of collaboration that are prevalent on the Web. The "Data Visualization" section of the Digital Research Tools Wiki at www.digitalresearchtools.pbworks.com/Data+Visualization identifies a variety of tools to "help users discern patterns in data"; many additional resources may be located by searching broadly outside the humanities, in fields such as the physical, social, and life sciences which have well-established data visualization toolkits. For communally-created hypertexts, evolving through thousands of rapidly-published versions created by hundreds of authors, observation of patterns rather than character-by-character comparisons is a necessary first step in identifying specific targets deserving of detailed study. Like computer software and living cells, hypertextual ultrastructures are best studied with the aid of visualization tools to bring them into a range compatible with our sensory and intellectual processes.

In addition to building on existing work in data visualization, my project is related to but different from Matthew G. Kirschenbaum's *Mechanisms: New Media and the Forensic Imagination*, in which he investigates electronic texts at the hardware level, as physical inscriptions in magnetic and optical storage media; Kirschenbaum does not use the term "ultrastructure," preferring terms from nanotechnology as more suited to the physical nature of storage devices. Our studies share a goal of demonstrating that what Kirschenbaum calls "screen essentialism" (19) is a watery and inadequate approach to studying digital texts: what the user experiences on a computer (or television or telephone or any other) screen is just the very thin surface layer of a deeply-structured text. My project is located in the space between the screen and the hardware, in the software-controlled realm that transforms the binary digits stored in hardware into the words, images, and sounds experienced by human users; I will draw examples from the surface level of the user's experience at a screen, but the purpose of those examples is to show the result of sub-surface software activities. The hardware and software that create and limit the user experience of computer-mediated textuality are as essential as paper (the storage medium) and ink (the method of marks to be stored) are to the study of print-based textuality; they are also similarly under-studied.

What Are Printed Texts Made of, and Who Studies Ink?

Studies of the material book have yielded detailed information about the social and industrial processes historically involved in print production, and the ways in which printing and culture have interacted and combined to create "print culture"; important works in this vein include Elizabeth L. Eisenstein's 1979 *The Printing Press as an Agent* *of Change*, Bruce Michelson's 2006 *Printer's Devil*, as well as the "Studies in Print Culture and the History of the Book" series by University of Massachusetts Press and the "Studies in Book and Print Culture" series by University of Toronto Press. There is a large and growing collection of information about print culture at the level of the book itself and at the levels in the cultural hierarchy at which books are aggregated: libraries, booksellers, and publishing enterprises are all being thoroughly studied. At levels below the finished book—the printed book's building blocks, such as ink, paper, and binding—textual scholars have had much less to say. This may be because the study of books' components, rather than books as finished products, is likely to require skills that exist outside the humanities; chemists, for example, are more likely than textual scholars to have the training to determine and discuss how ink and paper operate, and what physical possibilities can be opened or closed when alternative varieties of those components are paired.

Although chemists are best prepared to understand the low-level structures and processes defining the materials of which books are constructed, the work of textual scholars can be powerfully affected by the physical limitations of book-construction materials such as ink and paper, which are ordinarily below their radar. An important example of this is Ad Stijnman's 2002 report on "Iron-Gall Ink and Ink Corrosion," pointing out that, as the most widely-used ink from the 10^{th} to the 20^{th} centuries in Europe and European colonies has been found to corrode paper and parchment, "the production of a millennium" (172) is at risk for damage or destruction. Irreplaceable documents from previous centuries were written with ink that will destroy them; in this case, understanding why and how the destruction occurs may not lead to discovery of a method to prevent the physical damage, but it should lead to appropriate mitigating actions. For example, within the many large collections of documents which are candidates for eventual digitization, the scanning schedule could be adjusted so that the documents written with iron-gall ink are scanned first since they are at great risk of corrosive damage caused by that ink. This means arranging work in response to the physical components of a document, digitizing the most structurally-unstable documents first rather than those most interesting to current scholarly inquiries.

What other decisions about their own work could textual scholars adjust if their attention included the structural characteristics of the documents they study? If the items indexed by the *MLA International Bibliography* are a reasonable indication of scholarly interest in the matter, we are not likely to know the answer to this anytime soon. A November 23, 2007 search for items with "ink" in the title returned 131 hits; for most of these, the word was used metaphorically, leaving 24 items that were actually *about* ink.

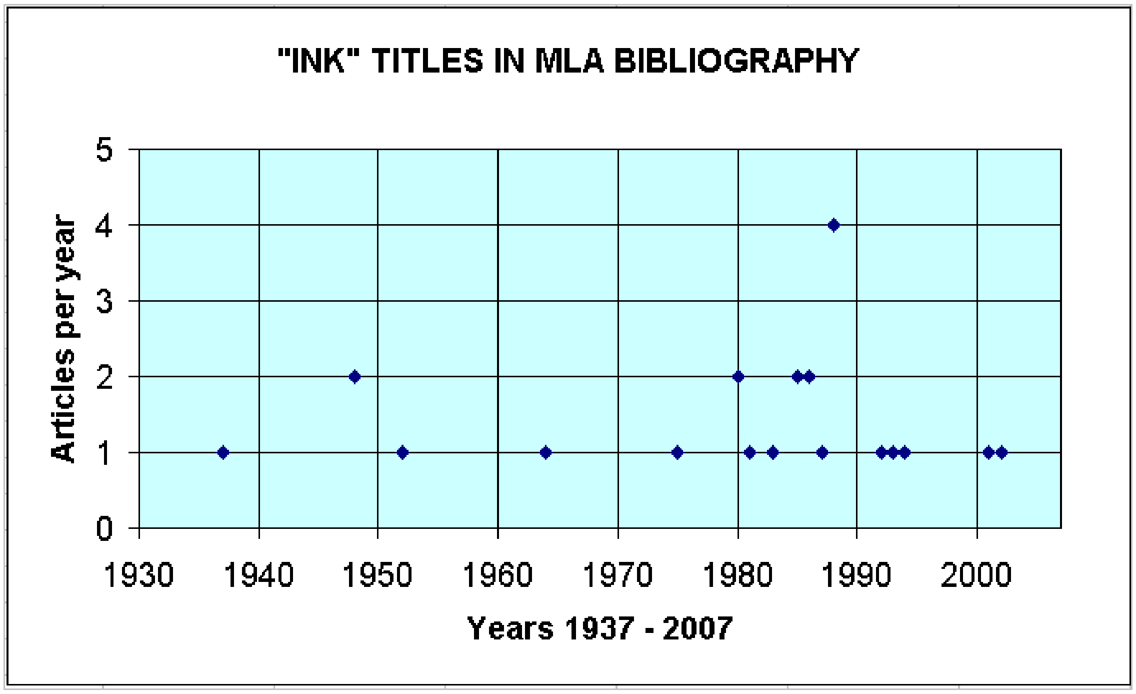


Figure 1: Frequency of MLA-indexed articles on "ink" 1937-2007.

As shown in Figure 1, there was a small flurry of interest in ink in the 1980s, peaking in 1988 with four articles published in the Spring-Summer 1988 issue of *Literary Research*. Otherwise, there has almost never been more than one article per year; in each of the 1930s, 1950s, 1960s, and 1970s, there was one article with “ink” in the title for the entire decade. For the years since Web-based publication has become possible, Stijnman’s discussion of ink corrosion points out an important consideration in assessing the intensity of interest in low-level physical components such as ink: the InkCor Project, sponsored by the European Union, has its own Web site at www.infosrvr.nuk.uni-lj.si/jana/Inkcor; scholars who are addressing the problems caused by ink corrosion exchange information there and at the Ink Corrosion Website at www.knaw.nl/ecpa/ink (sponsored by the Royal Netherlands Academy of Arts and Sciences) rather than publishing their work in the journals indexed by the *MLA International Bibliography*. These Web sites are indexed by general-purpose search engines such as Google, rather than by special-purpose indexes such as *MLA International Bibliography*; participants in the discussion there may be specialists and scholars, but their expertise is multidisciplinary and their contributions can be examined without the need to have joined a library which has purchased a subscription to the *MLA International Bibliography*. The Modern Language Association's "Frequently Asked Questions" Web page says "Subscriptions are offered to libraries through four vendors: EBSCO, Gale Group, OCLC, and ProQuest-CSA. If you have questions about whether or not your institution subscribes, ask your reference librarian for assistance"; those not affiliated with a good library will get no assistance from "*the* research tool for literature" (www.mla.org/bib_faq). In contrast, the list of 337 items indexed by the Royal Netherlands Academy of Arts and Sciences as relevant to the study of ink corrosion is available to all, as is the full text of some of the articles; the indexed articles were originally published in fields such as chemistry (*Journal of Chromatography*), library science (*Library Conservation*), and art history (*Journal of American Institute of Conservation*). There is an active and public international effort to study ink as one of the fundamental building blocks of early manuscripts that, as it decays and fails, is un-making the manuscripts it built; however, that effort to study a failing component and recover from its damage is happening at several centuries' distance from that component's design and implementation. Just as paper publications can be lost as their ink corrodes, films can be lost due to the deterioration of filmstock, especially what the National Film Preservation Foundation at www.filmpreservation.org identifies as "orphans," perceived as without commercial value and therefore unprotected by commercial studios' archival projects: the "documentaries, silent-era films, avant-garde works, ethnic films, newsreels, home movies, and independent works" of more than a century are decaying as their recording medium fails. For digital hypertexts, built of components designed in the past twenty years, beginning now to critically examine their deep nature can make it possible to prepare for and prevent the damage that will come when their components are also inevitably found to fail.

Paper Is to Screen as Ink Is to Software

For the same years as the "ink" publications charted above in Figure 1, searches for discussion of the elements used in hypertext construction also return sparse results. A hypertext is a readable work, analogous to a book or an essay or some other designation of a complete textual product; as a printed text is composed of identifiable components such as ink, paper, and binding, so are hypertexts constructed of lower-level components such as markup (expressed in markup languages such as HTML), database queries (expressed in query languages such as SQL), and scripts (expressed in programming languages such as Java and PHP). A November 2007 search for the most-popularized of these, "HTML," in the title of works indexed by the MLA International Bibliography yields only the five items listed in Table 1.

Table 1: List of MLA-indexed articles on "HTML" 1937-2007.

Year	Journal/Book Title	Article Title
2004	*Writing Instructor*	Learning to Love the Code: HTML as a Tool in the Writing Classroom
2000	*Grazer Linguistische Studien*	WEGE2000-ein Internet-taugliches Programm in HTML/JAVA-Sprachtraining im System (An Internet-Suited Program in HTML/JAVA Speech Training in the System)
1999	*Readerly/Writerly Texts: Essays on Literature, Literary/Textual Criticism, and Pedagogy*	Teachers Learning (Not Teaching) HTML with Students: An Experimental Lesson Plan for Introducing Web Authoring in Writing Classes
1999	*Computers and the Humanities*	An SGML/HTML Electronic Thesis and Dissertation Library
1999	*Computers and Composition: An International Journal for Teachers of Writing*	Reading between the Code: The Teaching of HTML and the Displacement of Writing Instruction

Of the five articles, four discuss pedagogy, using HTML as a tool to teach writing skills; the other reports on University Microfilms International's initial experiments in storing dissertations in Portable Document Format rather than as microfilm. For XML, the ink-like markup language most widely used in electronic publishing and digital archive projects, the trend is similar: of twenty-one items with "XML" in their titles, five discuss its use as a teaching tool; three describe uses in lexicography, and two point out implications in technical writing. XML and HTML are both derivatives of SGML, related to it as French and Italian are related to Latin, if Latin had survived as a spoken national language as well as a source for other languages. The *MLA International Bibliography* has indexed eighteen articles with "SGML" in their titles, fifteen of them published before the year 2000, one as early as 1987; the early low level of interest in SGML has become the current low level of interest in XML and

HTML. As shown in Figure 2, there has never been a year in which more than seven MLA-indexed articles referred in their title to any of these widely-used markup languages. In fact, the low numbers in Figure 2 are slightly inflated: the 1999 article on development of an electronic dissertation library, listed above as one of five HTML-titled articles, is counted twice because its title also mentions SGML; "Les langage SGML es ses applications en terminologie" by Jean Fontaine is included as both a 1993 book article (in *Proceedings of the XVth International Congress of Linguists*) and a 1994 journal article (in *Initiales*).

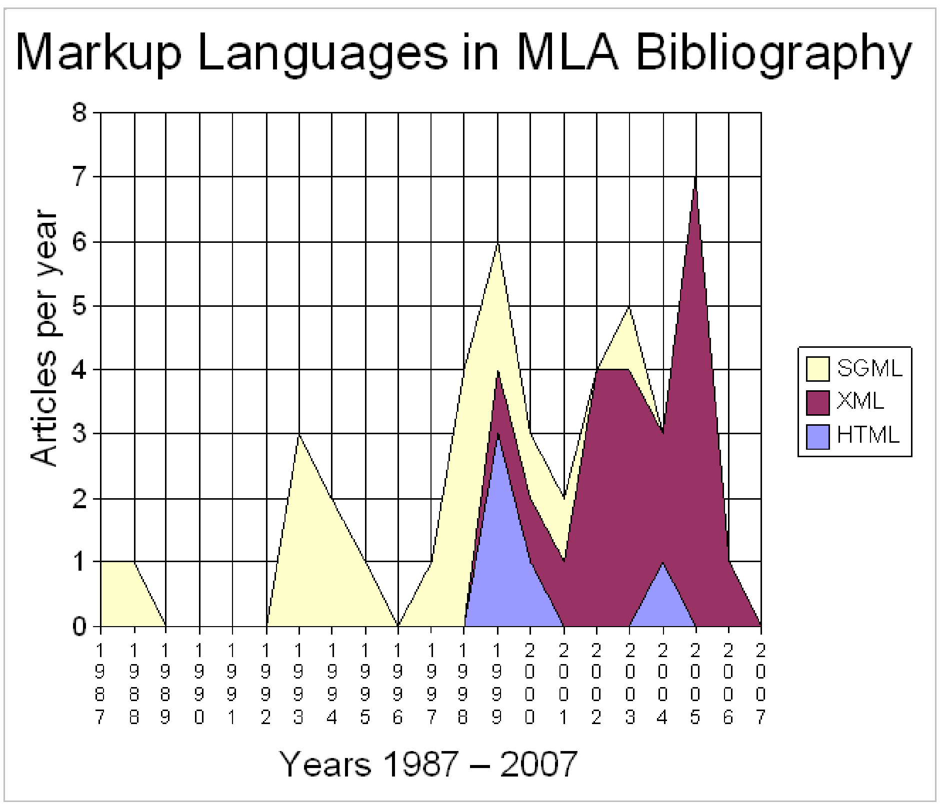


Figure 2: Frequency of MLA-indexed articles on markup languages 1987-2007.

True to the trend shown for "ink" articles, markup languages as key enabling technologies for hypertexts have received only minor published attention from the scholars who investigate other aspects of those texts.

For the same period, *MLA International Bibliography* indexes 338 items mentioning "hypertext" in their titles, the oldest being Kathleen Collins' 1988 "Hypertext: Baroque Ostentation and the Epigrammatic Speaker in the Salon Verse of Denis Sanguin de Saint-Pavin (1595-1670)" in *Papers on French Seventeenth Century Literature*; Tim Berners-Lee's first proposal for an HTML standard is dated 1989, so this and other very early discussions of hypertext in the late 1980s and early 1990s relate to what was at the time a completely experimental technology. Within the humanities, interest in hypertext, the human-oriented product of markup languages such as HTML, understandably started sooner and has been more widespread than interest in the markup languages themselves.

Of course, like ink, HTML is much more widely studied than could be surmised from the *MLA International Bibliography*: the Association for Computing Machinery's *ACM Digital Library* tags 38,057 articles as "about" HTML. Contributors to and readers of the sources indexed by the *ACM Digital Library* are likely to be specialists in the development of computer software and hardware. These are the people who invent the tools—languages, standards, architectures—that make it possible to publish digital texts.

If a digital publication is analogous to a printed publication, then the ACM's members are analogous to the inventors and improvers of paper, ink, and glue. As print culture developed and expanded, those inventors may have kept records of their own work—ideas, formulae, experimental results, debates among colleagues—but it is more likely that they followed the secretive standards of their guilds, protecting knowledge within a workshop and passing it orally from master to apprentice; if there are written records of the early work of such inventors, we have not found many of them. In our own time, as digital textuality develops and expands, the inventors of its enabling technologies also follow the standards of their "guilds": they publish prolifically, creating extensive records of their design intentions, change histories, and problem-solving attempts. As I will discuss in Chapter III, software is a technology built of words and wrapped in layers of other words; experts in words may be able to offer perspectives unavailable to experts in machines.

What Are Digital Texts Made of?

Web pages exist within and are made accessible by multiple layers of structured interconnection, described and standardized by the International Standards Organization (ISO) as the Open Systems Interconnection (OSI) Reference Model; the seven layers of the OSI model are labeled physical, data link, network, transport, session, presentation, and application (Cisco Systems). Of the model's seven layers, the presentation layer, just below the user-oriented application layer, is most relevant to this project. The presentation layer addresses portability of images, audio, and video, as well as standard coding and conversion functions that ensure cross-system and cross-platform readability of characters. In the problematic texts I will examine, errors occur not because presentation layer standards have been violated but because, although data is correctly transported between application layers of multiple computers, it is damaged by being handled by inappropriate methods or placed into inappropriate containers.

"Software" is itself a multi-layered element of multi-layered computer-mediated textuality. For some digital documents, notably hypertexts, much of the textual ultrastructure analogous to the OSI presentation layer is accessible to general users with the aid of very basic instrumentation such as the View—Page Source feature of a Web browser. Word processing software is also likely to include a method of revealing the ordinarily-invisible formatting codes that control a document's behavior. Other ultrastructural elements can only be observed via the tools and skills of a webmaster.

Paper Is (Relatively) Stable, but the Web Rots: Web Pages Require Caretakers

My project here is not to argue with Wimsatt and Beardley's contention that the proper way to study a text is on its independent merits *as a text* rather than as some expression of authorial intention or historical context or cultural voice. For printed texts, acting as if the author is dead may be useful and appropriate; for digital hypertexts, long-term separation from human supervision and maintenance is a sure way to cause the text itself to rot. "Rot" in the case of hypertexts is similar to the processes of decay in living organisms: usability degrades, structural integrity fails, identity is eventually lost. Protecting Web pages from this kind of loss, one task of the webmaster, is related to some of the tasks performed for printed texts in the same way that hypertext itself is related to text: recognizable as the same thing in its essence, but *hyper*-, more so.

Validity of a printed informational text may be carefully established before its publication, perhaps by combined iterative efforts of author, editor, proofreader, and fact-checker; with the facts established, the text is published and the validation effort is completed. For a Web page, though, validation is an effort that must continue throughout its existence: not only can the previously-validated facts of its linguistic content change, but all elements of its infrastructure can fail. My own experience as a webmaster, for instance, includes not only repairing hyperlinks accidentally broken when an external site changed, and reconfiguring software to block intentional damage by hackers, but rebuilding Web sites after their server's hosting company suddenly went out of business, making Web-published material suddenly unavailable anywhere outside my own backup copies. Again, a comparison to printed text illustrates: destruction of a Web server relates to the Web pages stored on the server somewhat as burning down a library relates to the printed texts stored in the library, but *hyper*-, more so: the parallel to printed texts would be more accurate if it were possible to simultaneously destroy all printed copies of the texts, in every library, bookstore, or private collection world-wide in which they might be found. Instant global access can also mean instant global loss; the webmaster's task is to prevent the loss where possible and recover from it when necessary, causing the Web page to reappear, globally available again as if all previously-destroyed copies of a printed text could be simultaneously restored.

The Task of the Webmaster: Ultrastructure from the Perspective of the Participant-Observer

"Webmaster" can seem a gendered term, and there are occasional practitioners who prefer the feminine "webmistress" or the neutral "webspinner." In this dissertation, I have chosen to use "webmaster" to refer to any person who has mastered (there being no equivalent “mistressed” in English) the complexities of creating and managing Web sites and who is engaged in doing so for one or more sites. My usage here will conflate and oversimplify what are actually separate specialties such as interface design, programming, database management, and performance analysis; because I will draw participant-observer examples from my own practice as sole proprietor of a technical communications agency, this conflation is actually a realistic depiction of my own experience and that of other solo practitioners.

What a webmaster actually does may not be readily apparent, even to experienced and skilful users of the Web. In talking recently with a potential client about a Web design project, I mentioned that I was quite busy “taking care of” about twenty Web sites for other clients; this was a startling new thought for him, that there might be some ongoing “taking care of” activity that persists after a site is created. As I explained at the time, this expectation of a text’s ongoing connection to human supervision is one of the ways in which publishing a document on the Web differs from publishing the same information in printed form.

The Web is in a permanent state of change; the texts of which it is comprised must change, too, if they are to remain usable. On the Web, the stable text is likely to be outdated, disconnected from current thinking and therefore invalid; the recently-updated text is likely to be maintained, actively attended to by a caretaker and therefore reliable. One of the most obvious ways in which webmasters maintain Web sites is by monitoring the health of their hyperlinks to other sites; if the linked-to page is renamed or deleted, or if its content is so altered that it is no longer an appropriate target, adjustments must be made. Other caretaking tasks are less obvious but require similar awareness of environmental changes: for example, a database structure that adequately supported ten visitors per day may be overwhelmed by a sudden surge in popularity, unable to respond to a thousand simultaneous inquiries; it is the webmaster's responsibility to develop means of adapting the site to this increase in workload. On the Web, increasing workload is generally perceived as a good thing: one writes in order to be read, after all. Because webmasters have access to useful details about the visitors to their sites (for example, what search terms led a visitor to a specific page), they are also routinely tasked with attracting additional visitors; this effort, often called "Search Engine Optimization" (SEO), can prompt continuous adjustments of the ultrastructural elements that describe Web pages to search engines so that the search engines can suggest the pages to human visitors.

In addition to the ongoing adjustments that maintain usability without altering a Web page's linguistic content, a webmaster may be called upon to implement changes to the human-readable elements of the page. For informational Web sites, this keeps the content of the site current, reflective of reality and therefore attractive to readers: news stories are updated as additional facts are discovered; government sites expand their frequently-asked-questions pages to address concerns raised by new legislation; commercial sites announce sale prices and clarify policies. This type of surface-level change, obvious to Web users while most other changes made by webmasters remain invisible, is often the basis for discussions of hypertextual instability and complaints about its social effects: if the return policy published today, after I have purchased a product, is different from the policy published yesterday, before I purchased a product, and may be different again tomorrow, when I decide to return the product, how is it possible to believe and act upon any information obtained from the Web?

The truth, of course, is that reality has always been unstable: events occur whether or not they invalidate a published Web page. Digital publication did not create instability, textual or otherwise, but it did encourage an expectation among readers that texts can be kept consistent with current information, and it improved the ability of writers to come close to meeting that expectation. Digitally creating and publishing informational texts increases the speed at which textual descriptions of reality change, bringing it closer to the speed at which reality itself changes. With this in mind, one of my interests in this dissertation is exploring how digital texts, including hypertexts, are marked with claims about their own stability and currency and validity, and what such claims have to say about the connections between reality and documentation.

Hypertexts as Real Texts

In concluding his 1992 *Hypertext*, George P. Landow summarizes the relationship between theory and hypertext as follows: “Contemporary theory can illuminate the design and implementation of hypertext, and hypertext in turn offers theory an empirical laboratory, a means of practice, refinement, and extension, a space, in other words, in which to test imaginings” (203). More than fifteen years later, it is difficult to imagine a worldview in which the primary value of the real is as a source of insight into the theoretical. The bizarreness of the observation is readily seen by substituting other elements in the positions of “theory” and “hypertext”: Clinical drug trials can shed light on the internal structure of humans, and human subjects of clinical drug trials can provide an empirical laboratory in which to test imaginings. True, but not valid: Humans don't exist for the purpose of testing the drugs that treat human diseases, any more than hypertext exists for the purpose of validating theory about hypertext. Reality, which includes humans and hypertexts, exists, whether or not any theory can explain it.

If a real text is one that a reader can obtain and use, and an imaginary text is one that, unseen, is suspected to exist safely in a library or museum, then a hypertext is the only real text of most documents that many readers, especially young readers, have ever seen and ever expect to see. In "What Can Psychoanalysis Tell us About Cyberspace?," Slavoj Žižek speaks of virtual reality as a fantasy world populated by "digital phantoms" (803), but hypertextual implementations of a wide variety of institutions (banks, libraries, shopping malls, government offices) and their publications are completely real for their users, providing many readers with their only method of access to the information around which they organize their lives.

Given that hypertextuality is a mundane aspect of the textual experience for many readers, it is no longer useful to pursue its study solely by pointing out hypertexts' differences from printed texts, explicating unusual examples of hypertexts, and arguing about the applicability of existing theories to hypertext. Instead, it is time to develop methods of describing, discussing, and debating hypertexts as we do printed and manuscript texts, but using tools and approaches designed to address the multi-layered nature of hypertextual construction and publication. Introducing and demonstrating several such tools and approaches is one purpose of this dissertation.

We do have extensive experience in analyzing printed and digital texts with features similar to hypertext, but this experience is concentrated in fields with major interest in works of *information*: technical communication, journalism, discourse analysis, software development. By experimenting with and refining tools developed for textual study of works of information, we can begin to adapt these tools to broader use, to include the works of imagination which are also coming to populate the Web and other places of hypertextual publication. At the same time, techniques developed for study and criticism of drama and film, in which imaginative ideas are contained within—and examined separately from—technological support and detailed performance instructions may be extremely appropriate for extension into the study of all forms of digital texts. Like software development, creation of a film can be a multi-year task involving hundreds or thousands of participants; methods of describing and evaluating the processes of collaborative authorship and revision in both these fields may be more applicable to studies of hypertextual authorship than any methods developed for the study of print-based publication. Studies of the material book are reminders that the container (scroll, codex, hypertext) in which ideas are packaged and distributed is worthy of examination on its own terms, independent of studies of the contents. We need ways of studying all forms of texts and all means of transmission between them, as well as concrete and granular methods of describing how the nature of a text can be shaped by its containers and the processes of movement between them.

Text, Hypertext, Move, Contain: Terminology and Notation

A central project of this dissertation is to explain and demonstrate what *ultrastructure* means as applied to hypertexts; the concept of ultrastructure is introduced above, in the “Layers and Ultrastructure: Hypertextuality, Inside Out” section of this chapter and is supported by examples and discussion throughout the dissertation. In lieu of a lengthy presentation of other relevant vocabulary, I briefly introduce here four additional key terms from the dissertation’s title. The definitions here are my own, for the purpose of illuminating their usage in the remainder of this dissertation; these terms may be used very differently in other contexts.

For the current purpose, a *text* is a coherent set of messages, structured as a single entity and intended, perhaps after multiple technologically-mediated steps, to be meaningful to a human. “Text” shares its etymology with “technology,” “textile,” and “architecture”: a text is necessarily a work of intentional construction. A hand-written grocery list is a text, as is a printed novel or newspaper, a celluloid film, an electronic mail message, or a wiki page. Some texts are digital, mediated by digital computing technology; in the preceding list, e-mail and wikis are digital.

Some digital texts have sub-surface layers of beyond-text features, making them digital *hypertexts*; texts of this kind are the primary focus of this project. Hyperlinking is one widely-studied feature of hypertexts: any text can include a suggestion that readers refer to a related text, but a hypertext provides a sub-surface connection so that, beyond simply suggesting a related text, it provides the reader with direct access to it. I argue that digital hypertexts can possess many other sub-surface features that distinguish them from other texts; identification and examination of the *hyper-* features of hypertexts will continue throughout the dissertation.

To *contain* a text is to hold it within its limits, maintaining a boundary between what it is and what it is not; a container is separate from its contents, which in the case of digital hypertexts are the surface-level messages available to human readers. A more traditional discussion of the ways technologies shape texts might relate differences among texts to differences in their publication media (comparing coverage of the "same" story in print and television news) rather than to differences in their containers (examining the "same" story in the blog from which it originated and the aggregator into which it was syndicated.) These are similar approaches, but they are not identical. Thinking about texts as existing within containers rather than as traveling through media makes it possible to imagine a published text as having an intrinsic nature and structure separate from the transmission processes it might experience: while a medium stands between and connects a sender and a receiver, a container surrounds the text itself; the sender and the receiver can interact with the container, potentially deforming or even replacing it, but the container's purpose relates to the text itself rather than to its transmission.

For a digital hypertext, to *move* from one container into another is generally a non-destructive process more like replication than relocation: the original text may continue to exist, inside its original container. Movement of text "from" one container into another can be metaphorical, in the same way movement of a human visitor "to" a Web site is metaphorical: the people, computers, and information remain in their original locations, but establish connections to and copies of each other as needed. The complications of what I call "structurally non-identical copies," created by moving a text's surface-level contents from one digital container (such as a word-processing document) into another (such as a Web page) are explored at length in the dissertation, especially in the chapters on error, versioning, and repackaging.

Several times in this dissertation, I include examples of artificial language which, while it resembles natural human language, serves the purpose of helping computers understand human ideas, ultimately in aid of a communication or publishing project meant to help humans understand each other. In such cases, the artificial language appears in a 10-point Arial font to visually distinguish it from the surrounding natural language. Fragments of markup languages (<table>), programming languages (echo), query languages (select), and command languages (Edit—Undo) are all represented in this way.

Authors of forum postings and software modules sometimes publicly identify themselves with their full personal or corporate names; alternatively, they may label themselves with aliases, nicknames, assumed names, account names, avatar names, screen names, user names, or other ways of distancing a public identity from a private identity. When I cite a source for which I think the author's full identity is not revealed in the author's published name, I mark that name by surrounding it with quotation marks: "sam07" and "otoor" are examples of this pattern; examples attributed to "rose" and "scribionics" are my own public postings. Similarly, a computer's directory structure can contain personal names. It is reasonable to assume that someone who stores and retrieves data in a directory named for a person actually is that person, but this is not necessarily so: computers and authorizations can be borrowed; one person can act as proxy for another, manipulating files on behalf of a supervisor or colleague. When my source for a person's name is only an observation of the file structure within which that person is working, I mark that name just as if it were their nickname posted in a forum: "Greg" and "Fatma" are examples of this pattern.

Ultrastructure Creates the Hyper- Attributes of Hypertext

Among the people who create and maintain them, computer systems are described in language that reflects a focus on containment and control: an operating system is protected within a *kernel*, accessible through a *shell*; applications connect via *pipes* and *sockets*; fire*walls*, breached by *tunnels*, enclose secure areas; storage is divided into *sectors*; privacy is ensured by en*crypt*ion. Although computers themselves operate within rigid structures such as these, the human experience of computer-mediated hypertextuality is routinely described in terms of unbounded exploration, endless interconnection, and openness to ongoing expansion and creativity. I argue that these seemingly-contradictory understandings of digital hypertextuality are reconciled by a deep structure which creates the *hyper*- attributes of hypertexts. This software-created internal structure operates like the ultrastructure of living cells, rigidly defining texts' nature and function. Hypertextual ultrastructure also resembles cellular ultrastructure in that it exists beyond the realm of normal human senses and cannot be interrogated without technological mediation; as with living cells, thorough study of deep structure is required for full description and useful analysis.

By exposing and examining the software-defined ultrastructures of sample hypertexts from mundane Web-based sources such as news services, government agencies, and shopping sites, I demonstrate concrete and granular methods by which hypertexts can be productively described, compared, and evaluated to address questions of interest to textual scholars: What errors are possible, and what are their causes and effects? How can evidence of revision be detected? How can multiple authors and their separate contributions be identified? How can a transmission history be established to relate multiple versions to each other? Additionally, I argue that this structural approach to criticism is necessary to a full understanding of informational texts just as it is for poetry and film, whether or not the text is mediated by a computer.

Each of the five central chapters of this dissertation is named to suggest, using language commonly encountered by computer users, one aspect of the complexities associated with computerized implementations of text. In part, I have named the chapters in this way because one of my projects in this dissertation is to examine what happens when an old text is placed into a new container. Computers are highly interactive and communicative containers, able to present warnings and suggest options in response to their own changing contents, so it is natural for me to organize my discussion of textual transition around what computers, as enabled by human software developers, have had to say about the process.

I am also motivated by my desire to demonstrate the vast areas of intersecting interest between information technologists and textual scholars. Development of specialized software tools to support textual projects has been and will continue to be valuable, of course, but I think the greater, synergistic, mutual value of “humanities computing” will arise when all parties recognize that the work they do independently, using the techniques and tools developed and approved within their own specialized fields, is in large measure the same, whether the object of study is an ancient manuscript or a modern Java script. Information technologists and textual scholars perform (and argue heatedly over) similar tasks such as identifying and correcting errors, tracking the effects of intentional change, placing ideas within appropriate supporting structures, and managing collaboration.

In selecting examples and case studies for close reading in this dissertation, I looked for texts that are representative of widely-occurring practices and problems; I am not seeking one-of-a-kind art objects, but demonstrations of larger patterns. I also chose examples that are recognizable as works of information rather than as works of imagination, in part because this is where my own professional interest and experience as a technical writer is focused, and in part because I found that, although hypertextuality in general and the Web in particular were developed to assist in managing information, textual scholarship has neglected works of information in favor of works of imagination. Primarily, though, I have chosen informational texts as examples in the remaining chapters because I think the techniques of presenting information lend themselves most readily to examination of the effects created by the container rather than the contents.

Chapter II: "Click Here to Agree"—Error and Validation, Claims and Disclaimers

In Chapter II, I discuss errors introduced in the process of transmitting an old text into a new container, specifically in terms of errors created by computerized processes such as optical character recognition and special character encoding. Efforts to minimize reader expectations of textual validity for electronic texts, as well as calls for reader assistance in identifying errors and rejections of responsibility for unnoticed errors, are also examined and compared to earlier attempts to establish the validity of printed texts. Brief examples in this chapter are drawn from customer-information Web sites provided by government agencies and commercial enterprises. Fuller case studies are developed from Google News (news.google.com) and from faculty Web pages for the Department of English at Texas A&M University; these case studies of informational Web pages have in common an exploration of possibilities for error when text is created (written) in one kind of electronic container but presented (read) in a different kind of electronic container.

Chapter III: "README"—Versioning and Comparison

Not all textual changes are due to error or loss in transmission to a new medium, and not all changes in the claims made about texts are due to uncertainty about textual validity. Chapter III examines established methods of documenting textual revision within the fields of textual studies and software development, both of which have long-standing goals of identifying and tracking textual instability. Much of this chapter consists of a case study of changes in the way contents are ordered and arranged when a text moves into a new container of the same type, comparing multiple print editions of a scholarly work on a rapidly-changing subject (George P. Landow's 1992 *Hypertext*, 1997 *Hypertext 2.0*, and 2006 *Hypertext 3.0*) and demonstrating techniques relevant to visualizing intentional structural change.

Chapter IV: "View Page Source"—Text, Tools, Code, and Metacode

Because this dissertation explores changes in textual structure, including the deep structure that creates hypertext, Chapter IV examines methods of causing and observing the unintended changes that can occur as texts move into and among digital containers. A case study of Web pages used for unintended purposes shows, from the perspective of the webmaster, how the post-publication uses of Web-published texts can be observed and responded to, and how their activity can require the webmaster to continuously monitor and adjust the container in which they were published. Additional examples identify methods of placing digital and non-digital texts in the correct relative chronological order, and methods of extracting meaning from "stray marks" within hypertexts.

Chapter V: "Edit This Page"—Collaboration and Control

Chapter V addresses issues of ownership and responsibility, and the overlapping activities of editing, writing, reading, and use, for texts which are communally produced and maintained. A case study of osCommerce, communally-developed open-source software, demonstrates an ultrastructural design which enables user preferences to shape not only the selection but the structure and content of dynamic Web pages. This demonstration is expanded into suggestions of ways to characterize Web pages as not only dynamic, but as composed of components that are dynamic in different and definable ways which can be aggregated into a detailed description of the nature and behavior of the page as determined by its ultrastructure.

Chapter VI: "Save As"—Repackaging, Repurposing, and Responsibility

Chapter VI examines ways in which movement of electronic texts into new containers can cause containers and contents to create, destroy, and alter each other. A case study from the *New York Times* compares a print-formatted front page to the digital archive's retrieval of the same stories, identifying ways in which each container limits or expands the ways in which the stories can be used. Another case study, of Ida B. Wells' 1894 *A Red Record,* demonstrates the applicability of visualization techniques for tracking the movement and reframing of an item of information, in this case a report of lynching, from its origin in the *Chicago Tribune* into the multiple printed and electronic media in which it has been repackaged.

Chapter VII: Conclusion

I conclude the dissertation by pointing out that, while the numbers and kinds of digital texts are currently rapidly expanding and show no signs of decline in popularity, hypertextuality is, like any form of technology, of a limited lifespan. Our current position, early enough in the development of this technology that we have full access to the details of its construction and operation, is the ideal moment for intensive study of hypertextual ultrastructures and the experiences they create for readers and writers. As time passes, low-level hypertextual components are likely to be merged into complex higher-level combinations, just as full-page laser imaging superseded moveable type;

now is the time ideal to examine and learn from the deep structures that create hypertextuality.

Toward a Formalist Criticism of Hypertext

This dissertation is a step toward a formalist criticism of digital hypertexts, demonstrating some of the observations, comparisons, and discussions that can be pursued when the necessary descriptive vocabulary and tools are available. If such a critical system is to develop into a truly useful way of thinking about hypertextuality, it will necessarily differ from established formalist approaches to literature and film in at least one major way: the author must retain a central place in the structure. Because, as I argue throughout this dissertation, computer-mediated hypertexts are so strongly shaped by and dependent upon ongoing human involvement, investigation of hypertextual forms cannot be accomplished without addressing the ways in which each form interacts with its caretakers and users.

The hypertextual forms I examine here—including news aggregators, blogs, wikis, electronic mail, forums, and static and dynamic Web pages—are ubiquitous at the time I am writing and so serve as a reasonable set of initial demonstrations of formal investigation, but this is hardly an exhaustive collection. While new hypertextual forms are being developed, it may not be possible to create for hypertextual studies the kinds of general-purpose reference material that support the formal study of poetry. It may be some years before it will be practical to thoroughly describe and explain “forms and genres major and minor, traditional and emergent” (vii) for hypertext in the way Preminger and Brogan’s *Princeton Encyclopedia of Poetry and Poetics* does for poetry.

A concise overview of the major forms of hypertext, in the style of Paul Fussell's *Poetic Meter and Poetic Form*, intended to help develop "an appropriately skilled audience of an exacting art" (xi) may be a nearer goal. Nearer still, it may already be possible to produce useful, formal studies that are narrowly focused within the limits of a specific time and technology: I imagine projects such as *Pre-Broadband Electronic Mail* or *Post-iPhone Web-Based Journalism*, if they were to be created, could be of immense value to current and future scholars attempting to understand recent steps in the evolution of communication and publication technologies.

In its own right and as a suggestion of areas deserving of further development, I hope that this dissertation will be of interest to several intersecting audiences. Textual scholars may find ideas about how to extend and adapt their current methods to both study born-digital texts and follow pre-digital texts into new digital versions. Technical communicators and software developers may prepare for new challenges in explaining how to use a system while having little awareness of the system's surface-level appearance. Rhetoricians may imagine how to appropriately set expectations for digital texts that exist as multiple directly-accessible components, usable independently of any initial disclaimers or concluding warnings. Journalists may address levels of meaning above and below the stories they write, with internal structures determining each story's placement in external syndications. Linguists, especially critical discourse analysts, may expand their studies of language in social power structures to include the human-controlled structures of digital hypertexts. Literary critics who wish to describe, discuss, and evaluate hypertexts as they do other texts—using formal, historical, biographical, and other methods—may discover new methods of doing so. Textual caretakers of all kinds—librarians, archivists, curators, webmasters—may find new arguments for the urgency of their tasks.

CHAPTER II

"CLICK HERE TO AGREE"—ERROR AND VALIDATION, CLAIMS AND DISCLAIMERS

Digital reproduction is reliable: this assumption is central to the popularization of digital audio recording, digital telephone and television transmission, digital signatures, and other computer-mediated methods of storing and publishing information. Barring damage from an electrical surge or some other rare physical disaster, digital transmission is highly reliable; digital copies are likely to be perfect, each identical to all others and to the original. Acting on this assumption, a writer has every reason for confidence that a digital text will reach readers intact, without the kinds of technological interference that have plagued printed communication. For printed texts, following the paths of errors in textual transmission has been a key method of identifying genetic relationships among texts; identifying and sometimes repairing those errors, giving readers access to an undamaged text of a work (or awareness of the ways in which the text has been damaged), has been a central project of scholarly editors. For digital texts, reliably transmitted as perfect copies, are either of these pursuits still necessary or even possible?

In this chapter, I argue that the digital technologies that create hypertexts also create errors, and that those errors are especially likely to occur at moments of transition from one kind of digital container into another, such as a transition from a private word-processing document into a public Web page. While words and images can be directly transmitted into new containers by digital means, the beyond-text internal structures which create hypertext must often be adapted or transformed rather than being copied, making them especially vulnerable to loss and damage; for hypertexts, I argue that efforts to identify, classify, and correct errors are properly focused at ultrastructure, which creates behavior, rather than at surface-level contents such as words and images.

The examples I will examine in this chapter demonstrate some types of errors that can be induced by transferring a digital text into a container for which it was not designed. Because Web-based search engines and other data aggregators create dynamic, single-use Web pages that attempt to combine digital texts designed for separate containers, many of my examples are drawn from these sources.

Boundaries, Membranes, Bindings

An object is distinguishable from other objects because it is physically bounded in some way, contained within some wrapper that separates what-it-is from what-it-is-not. For humans and other animals, that wrapper is our skin, the largest organ in our bodies, separating and protecting our mostly-liquid insides from the mostly-gaseous atmosphere and mostly-solid land surrounding us. Left uncontained, liquids diffuse and commingle, losing their separate identities; containment makes it possible for us to function as individuals. The skin which contains us is made of cells which, at the ultrastructural level, each have their own wrappers, each cell membrane "regulating the contents of the cell, for all nutrients entering the cell and all waste products or secretions leaving it must pass through this membrane. It hinders the entrance of certain substances and facilitates the entrance of others" (Vilee 41). Membranes define and contain us, but the container is permeable; materials must move in and out, or we could not live. Containers are so important to living things that the major classification of an organism as a plant or an animal is based on the nature of its cellular containers: if the cells are bounded by cell walls in addition to cell membranes, the organism is a plant.

Other objects, such as printed books, exist within other kinds of containers. A book is not a living being, so its container need not be physically permeable. Examination of the physical construction of a printed book is likely to reveal several types of container, reinforcing each others' claims about what the book is and how it is to be used. The appearance of the outer cover of the book, binding the book's pages within, makes some general claims about the nature of the book; for an experienced reader, a book's cover is almost certainly sufficient to set expectations appropriate to its identity as a cheerful children's book, a serious scholarly book, or a practical recipe book. The cover also displays a title which, properly interpreted in the light of its cultural context, makes specific claims about the subject matter of the book; for example, on the cover of José Eduardo Agualusa's *The Book of Chameleons*, the tiny note "A NOVEL" clarifies that, although the title and the cover art (a close-up of brightly-colored reptilian skin, with a white cut-out of a lizard's silhouette) might be appropriate for a work of herpetology or natural history, this is a work of fiction. Inside the cover, before the story begins, the copyright page repeats and extends that identification: "This book is a work of fiction. Names, characters, places, and incidents either are products of the author's imagination or are used fictitiously." Close examination of the cover art reveals that the lizard's silhouette includes one tiny human hand, confirming that this is a work of imagination.

Printed books are not the first or only containers of texts to carry markers identifying their contents. Papyrus scrolls were sometimes also marked with attached tags, “an ancient form of today’s Post-It note” summarizing the manuscript “to save readers from unnecessarily unraveling a long irrelevant document” (Langville and Meyer 1). Like manuscripts and printed texts, digital texts begin and end at marked boundaries and carry identifiers of their nature and intended use. When a digital text fails to behave in a way consistent with its label and container, it fails in the same way its mediating software can fail, not necessarily by not functioning at all but by not functioning as its designer intended, even if the “designer” is an automated aggregator, re-packaging the text into containers its original human designer never imagined.

Examining the containers of digital texts, searching for their mechanisms of identifying the nature and purpose of their contents, makes it possible to recognize when a digital text is failing to achieve its stated purpose (experiencing an error), which then makes it reasonable to investigate whether that error has its source in external causes (damage to the database, for example) or internal structures (a design incapable of meeting its goals).

Texts Wrapped in Claims

Texts make claims about themselves, guiding readers as to how to interpret the text before it is read. To a large extent, texts’ claims are based in and expressed by their containers. A news story, for example, is presumed to be factual rather than fictional because it is contained within a newspaper; newspapers sometimes contain elements—crossword puzzles and comic strips—that are meant not to inform but to entertain, but these elements are expected to be clearly marked as non-news, not normative within the larger container of facts. As a news story moves from a container with well-established claims (such as a printed newspaper, with polished prose, a consistent house style, and carefully-selected images) into a container with more ambiguous claims (such as a Web-based newspaper, with frequently-updated news stories intermixed with reader-contributed commentary and phone-cam video), that movement affects not only the text but the claims that can be made about the text.

Texts in media such as print and film are relatively easy to mark with obvious and indelible claims of category and ownership; the claim markers can be ripped away, but probably not without noticeable damage to the surrounding text. Digital hypertexts are less firmly tied to covers, copyright pages, and other claim markers; Web pages, especially, are so prone to fragmentation, separation from original context, and adaptation to environmental pressures that it can be difficult to establish reasonable claims about how the page should be categorized and what limits to its use may be appropriate.

Texts Inconsistent with Claims: Error

Because hypertexts are interconnected with each other, with all the connected texts changing independently, the claims made about a hypertext from within may contradict claims about the same hypertext made from without. Figure 3 demonstrates some of the complications that can arise.

Figure 3: Google News claims about a Voice of America story.

In this example, *Google News* at news.google.com collects stories from news Web sites and claims to update its collection frequently ("Auto-generated 21 minutes ago"); it also claims to have collected a news story from the *Voice of America* (*VOA*) site "1 hour ago" (presumably, one hour prior to the auto-generation of the *Google News* page 21 minutes ago), for which the headline is “Hamas Claims Responsibility for Jerusalem Shooting.” By providing a hyperlink connecting that headline to a news story, *Google News* claims that the reader will be able to read the headlined story. Clicking on that hyperlinked headline does present a *VOA* news story about a shooting in Jerusalem, but that story begins with a different headline (“Thousands Mourn Victims of Jerusalem School Shooting”) and its content (“Hamas is denying earlier claims that it was responsible...”) contradicts the headline as shown in *Google News*. Because the *VOA* story claims currency by marking itself with a day (“07 March 2008”) of publication while the *Google News* site tracks stories’ currency by minutes, it isn’t possible to precisely relate the timing of *VOA*’s changes of its headline and story content to the timing of *Google News*’ collection. Re-visited two days later, the *VOA* story at www.voanews.com/english/2008-03-07-voa26.cfm has the same headline (“Thousands Mourn Victims of Jerusalem School Shooting”), the same dateline (“07 March 2008”) and most of the same content: what *Google News* presented as “Hamas is claiming responsibility for a school shooting” (Figure 3, top) and hyperlinked to *VOA*’s revised “Hamas is denying earlier claims that it was responsible” (Figure 3, bottom) has become “Hamas praised the attack, calling it revenge for Israel's recent offensive against militants in the Gaza Strip, but stopped short of claiming responsibility.”

The Google site makes a claim that the story at the *VOA* site contains certain text; at the *VOA* site, the promised text is not there. It must once have been there, when Google collected its information, only to be changed before the story could be read by following the hyperlink from Google. A reader regularly checking the *VOA* site without the mediation of a *Google News* hyperlink would not be disturbed by multiple revisions of the news story; after all, providing the latest information about breaking news, even when it contradicts earlier information, is in the tradition of good radio journalism as practiced by *Voice of America* since 1942. The tradition is complicated, though, by *Google News*' extraction of stories on a schedule different from the *VOA* journalist's ongoing efforts to keep the story current.

Within the container of a radio broadcast, a news story might be re-read every hour with the latest revisions in place; previous versions of the story might be remembered by listeners, but all but the most-recently-read version would be mentally discarded as obsolete. *VOA*'s Web site, though, is more like a printed newspaper, with all stories permanently available to readers, than it is like a radio broadcast, with each story available only for a few minutes while it is being read aloud. Within the tradition of print journalism, "the purpose of the headline is to urge the reader into the story" (Baskette and Sissors 137). Believing that the headline in *Google News* must accurately summarize *VOA*'s story, a reader who came to the *VOA* site by following the hyperlinked headline from *Google News* would be surprised to see the story not as promised. If, as in this case, a hypertext describes and hyperlinks to another hypertext which does not match the description (although it may once have done so), is that an error? If it is an error, what kind of error is it (is this a software bug, an invalid configuration, a simple accident?) and where does responsibility for that error reside: in the linked-to hypertext that changed, or in the linked-from hypertext that did not incorporate the change, or in the Web that allows hypertexts to connect to each other superficially, without providing synchronization?

Studies of the material book have developed beyond the conflation of all possible typesetting problems under the single heading of "incidental errors," making it possible to ask (and sometimes to answer) questions requiring a detailed understanding of the production process: Did the typesetter accidentally choose type from the wrong case (and what does that say about typesetting's place on the scale between careful craft and mindless mass-production)? Did the typesetter intentionally substitute a similar-looking letter "V" in response to a shortage of the correct letter "U" (and does this mean anything about general attitudes toward accurate reproduction)? Was the correct-but-poorly-made letter worn down in the course of a long printing run, leaving late impressions with an incorrect "P" rather than the correct "R" of early impressions (and was this because profit margins were too tight to support high-quality type)? Was an unexpected blank spot caused by uneven application of ink, or by faulty installation of type, or by wrinkled paper (and what can be learned about the development of quality control as an ideal in the publication industry or in the general culture)? With hypertextual ultrastructure as a lens with which to examine how hypertexts are constructed, it becomes possible to develop a similar level of detailed critical understanding of what can go wrong and what must go right in order for a hypertext to operate correctly.

Types of Error

In "Nonlinearity and Literary Theory," Espen J. Aarseth identifies several ways in which a text can change in the process of metamorphosis from one medium into another:

> unintentional (the blunders of a typesetter or projectionist in the dark), usurpatory (a re-mix of samples from a musical recording, a hacked version of a computer game), plagiary (one composer's unacknowledged variations on the themes of another) and subversive or estranging (the "cut up" textual experiments of William Burroughs and John Cage), to suggest a few. (57)

Here, Aarseth lists three categories of intentional changes (usurpatory, plagiary, subversive), leaving all forms of unintentional changes in the category of "blunders." My interest is in a different kind of change, change which is imposed by the nature of the text's new container rather than by the intention of the person—author or editor or publisher—who decided that the text should move into that container. Such changes can be categorized as intentional, since movement into a new textual container does not occur accidentally, but the intentional movement exposes the relocated text to unintentional damage far more subtle than can be fairly described as simple "blunders."

For example, in "Orthographic Errors in Web Pages: Toward Cleaner Web Corpora," Ringlstetter et al. examine a large sample of English-language and German-language Web pages to assess the validity of the Web as a repository of natural language for linguistics research, finding "a non-negligible number of orthographic and grammatical errors occur in Web documents" (295). They categorized these errors according to four apparent causes: *typing*, *spelling*, *encoding*, and *recognition*. Typing errors, analogous to Aarseth's "blunders," are caused by physical misplacement of a typist's hands on a keyboard, accidentally substituting a nearby character for the intended character, although the typist knows how to spell the word and intends to do so correctly. Spelling errors, caused by mental rather than physical mistakes, are recognizable when a word is consistently misspelled on a page, showing that the typist does not know how to spell the word and has produced it on the page as an accurate representation of that lack of knowledge. Encoding errors are caused not by a typist but by software; if the text could be examined independently of the software that formats and presents it as a Web page, the text itself would be judged to be correct. Recognition errors can be caused by the software or hardware used in creating a digital text from another form; Ringlstetter et al. mention only Optical Character Recognition (OCR), used in scanning paper documents to create digital documents, as a possible source of such errors, but since similar processes can transform sound into text or text into sound, I prefer to generalize the category to include all manner of recognition errors.

Some of the observations in Ringlstetter et al. deserve additional investigation, but are beyond the scope of this dissertation other than as examples of additional types of errors that must be accounted for on the Web. For example, they report that English-language Web pages are the most likely to be error-ridden. I suggest that in part this occurs because English on the Web is widely produced by non-native speakers who have minimal proficiency in the language but must use it to participate in online discussions, while a Web page in any other language is likely to have been produced by a native speaker of that language and to be intended for readers who are also highly proficient in that language. I think the high rate of errors in English pages is also partly explained by the use of English as the basis of programming languages and of the telegraphic expressions, commonly "texted" on telephone keypads, which have become common in other media. For example, in a discussion forum for programmers working with the osCommerce open-source shopping cart software, "otoor" seeks help at www.forums.oscommerce.com/index.php?showtopic=316523 in setting up a system to use the Saudi Real as currency, receiving technical advice on this from "web-project" in the UK and "Bob K" in Malaysia. In the middle of the conversation, the errors in one statement by "otoor" demonstrate both possibilities:

> thanks 4 ur post.
>
> what is "currency contribution" ?! and where I can found it and how to add it in my store?

On the tiny screens and tiny keypads of telephones, "4 ur" routinely replaces "for your"; this is an intentional substitution to improve efficiency, not ignorance of the correct spellings and not an accidental slip to the wrong keys. If this is an error, it is in creating a text for one container (the Web, which is usually—but not always—displayed on large desktop screens) by a method—telegraphic typing—designed for another container (tiny handheld telephone screens). There is no effort to save space in the remainder of the online conversation, as a large snippet of code is posted and space-consuming visual techniques (bolding, blank lines, large font) are used to make the text easier to read. In the context of the entire discussion, “4 ur” is a mismatch, perhaps what I will discuss later as a “containment error,” only problematic because it does not meet the expectations of the full-size Web page on which it is displayed. Other errors by “otoor” (“where I can found it” rather than “where can I find it”) really are signs of linguistic difficulty; the task which “otoor” discusses in the forum, that of creating an online shop which uses the national currency of Saudi Arabia, suggests that this online discussion was conducted in English because English provides access to the global community of programmers, not because English is the language in which “otoor” best communicates.

Errors of typing (creating an unintended result) and spelling (creating an incorrectly-intended result) are certainly not unique to digital documents. Neither are errors of recognition: a human listener can fail in the same ways a computerized recognition device can, misreading a written character or mis-hearing a spoken word, especially in cross-cultural or noisy circumstances. Encoding errors, though, are uniquely relevant to digital documents, especially to hypertexts, due to hypertexts’ ultrastructure of multiple layers of containment.

I divide encoding errors into several sub-categories; for example, linking errors occur when a hypertext is intended to provide the reader with a hyperlinked connection to another hypertext but fails to do so, either because the target location is not where it was expected to be (an incorrect address is correctly expressed) or because the path to that location has not been correctly specified (a potentially-correct address is incorrectly expressed). Linking errors are pervasive in Web pages, especially when a single page is moved up or down within the multi-level directory structure of a Web site. When a Web page is placed in a different directory, it can acquire linking errors that did not exist in its original location, and that will cease to exist if it is returned there. This occurs when hyperlinks follow a relative path that assumes a known starting point at a fixed distance from the endpoint ("go up three levels from here, then open Page X") rather than an addressing an absolute location as the endpoint. In such a case, examination of the Web page's ultrastructure shows that what behaves as a linking error is actually a kind of containment error: the hyperlink probably worked correctly in an older copy of the Web page which was stored at a different location, and would probably work again, unchanged, if the Web page were copied back to that location.

Figure 4 relates my ideas about containment errors and other encoding errors to Ringlstetter's ideas about Web errors and Aarseth's ideas about textual change. Since all change must be either intentional or unintentional, Aarseth's initial division includes all possibilities. Within those categories, though, there may be more subcategories than identified by Aarseth or Ringlstetter; as I demonstrate below, there are definitely additional useful subdivisions of the encoding errors that occur in Web documents.

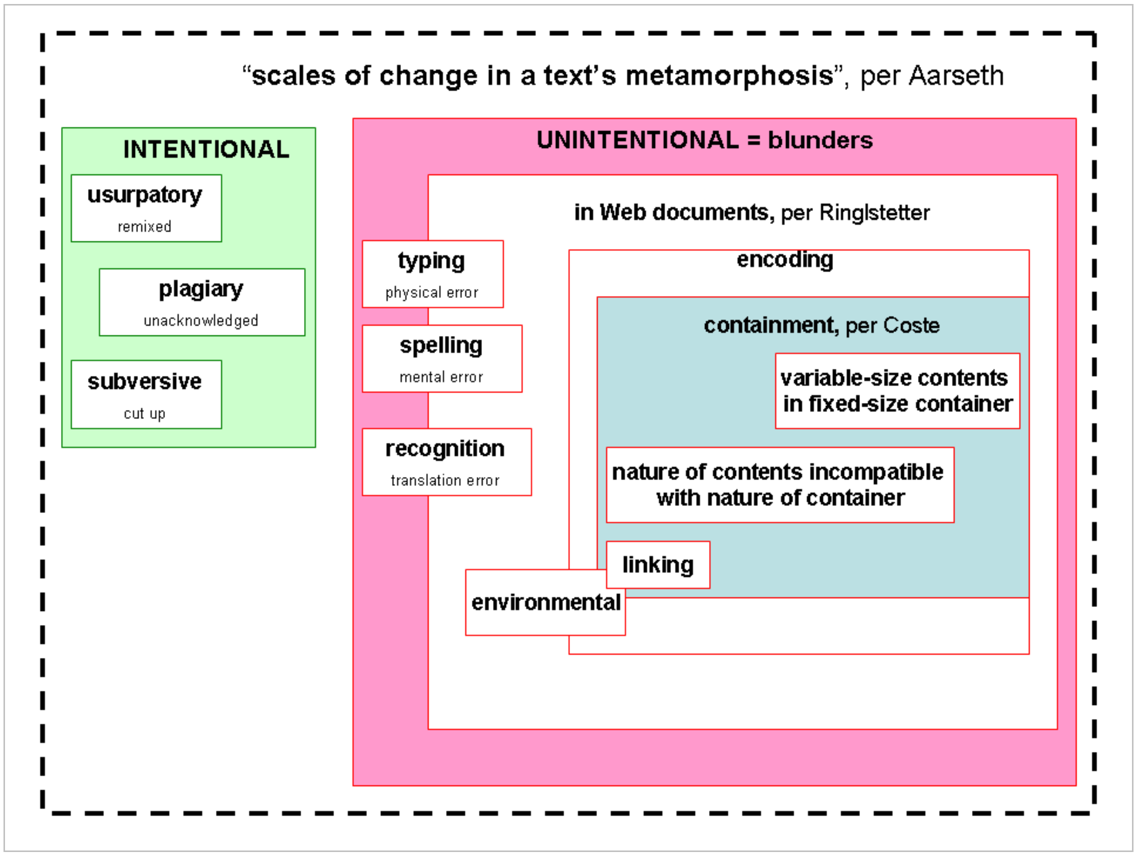


Figure 4: Relating three models of errors introduced as texts change.

Types of Containment Error

Some textual containers are open and adjustable. A handwritten grocery list, for example, can exist with complete validity on one side of a sheet of paper, or on both sides of the same sheet of paper, or on one or both sides of many sheets of paper; the grocery list expands as more pages are added, or contracts as sheets are lost or discarded, but any portion of it is usable as a grocery list and all portions can be expanded and contracted without losing their identity and usability as a grocery list. Some electronic documents also operate as open containers. Word processing tools such as Microsoft

Word create electronic documents that expand and contract even more fluidly than a paper-based document can: add words, change fonts, or shrink the size of an individual page, and the Word document adds pages as needed to fit the contents; delete words, or increase the size of individual pages, and any now-unnecessary pages are dutifully deleted, with the document remaining recognizable and usable no matter how the size of its container shifts.

Other textual containers are not so flexible: the container's attributes are fixed and, if there is any adjustment to be made in the interest of compatibility between container and contents, that adjustment must be made by the contents. The tightly-structured nature of hypertext makes it especially prone to errors of this nature; when a hypertext is designed for a container with a specific set of attributes, but displayed in a container with different attributes, the result is what I have termed a "containment error," which I identify as a special type of encoding error. In the case of a containment error, the hypertext is built correctly, in that all the text is available and all the hyperlinks are functional, but the hypertext does not operate correctly because it is incompatible with its container. In what follows, two examples illustrate two kinds of containment errors.

Examples: Size of Contents Incompatible with Size of Container

A visit to the United States Patent and Trademark Office (USPTO) Web site at www.uspto.gov reveals containment error caused by variable-size text in fixed-size containers; its home page is shown in Figure 5.

Figure 5: Containment error: variable-size text in fixed-size containers.

As has become customary for Web pages published in languages read left-to-right, the left side of the page is occupied by navigational controls which control the content of the central area of the page: for example, clicking on “Trademarks” provides a list of trademark-related hyperlinks, and clicking on one of those hyperlinks brings the relevant information into the browser window, replacing the home page display. On the home page, most of the navigational controls are clearly readable, but several are not: the second item, “Under Secretary &,” seems to be a truncated idea, and the next two items seem to have multiple lines of text overstruck.

The Web browser software which mediates reading of this page allows each reader to vary the size of its text, customizing it to accommodate the limitations of human eyesight and the capabilities of computer monitors; the HTML through which the Web page is defined, though, establishes fixed-size containers which do not adjust to changes in the size of the text they contain. Looking at the same section of the Web page, the navigational controls, seen at several text sizes using the same browser (Mozilla Firefox 2.0.0.9) on the same computer monitor (AOC liquid-crystal display at resolution 1440x900 pixels) shows that the text was designed to be viewed at only one size.

One “View—Text Size—Decrease” below the default browser setting allows all text to be read within its proper containers.	The default browser setting leaves some text truncated and other text overstruck and illegible.	One “View—Text Size—Increase” above the default setting forces some text has out of its containers.

Figure 6: Containment error: readability changes as user-selected text size changes.

As shown in Figure 6 and Figure 7, while the size of the text changes, the size of the text's visual container—the grey-shaded rectangle which serves as a background for the text—does not change. Investigation of this Web page's ultrastructure explains why this occurs: the elements combined to form the left-side navigational controls are incompatible. Hyperlinks are formed by anchoring the address of a new page to an element of the current page; that element can be an image, often made to appear as a three-dimensional "button," or it can be a string of text. In this case, each hyperlink is anchored to a string of text, and each string of text is placed over an image, with the pattern repeating for each of the thirteen hyperlinks offered. The gradations in the image's color suggest a vertical stack of slightly-curved buttons, with each button labeled with text to indicate its purpose; however, the foregrounded text label and backgrounded button image are separate items. Changing the size of the text changes the size of the label, but the image's size is fixed: the image is 162 pixels wide and 30 pixels high, no matter what size its companion text becomes.

With text size increased two steps above the default setting, “Products & Services” has become entangled with “Strategic Planning” and “How To.”	Increased three steps, much of the text is overstruck.	At six steps above the default setting, the text is no longer readable as navigation links.

Figure 7: Containment error: extreme increase in text size improves visibility but degrades readability.

This behavior of navigation-control text spilling out of its graphical container is not unique to the USPTO’s site. The “Welcome to the White House” page at www.whitehouse.gov uses a modification of this approach, adding text-only hyperlink anchors for left-side navigation. As shown in Figure 8, a containment error at this White House page means that viewing the page with enlarged text causes several of the navigation controls to disappear.

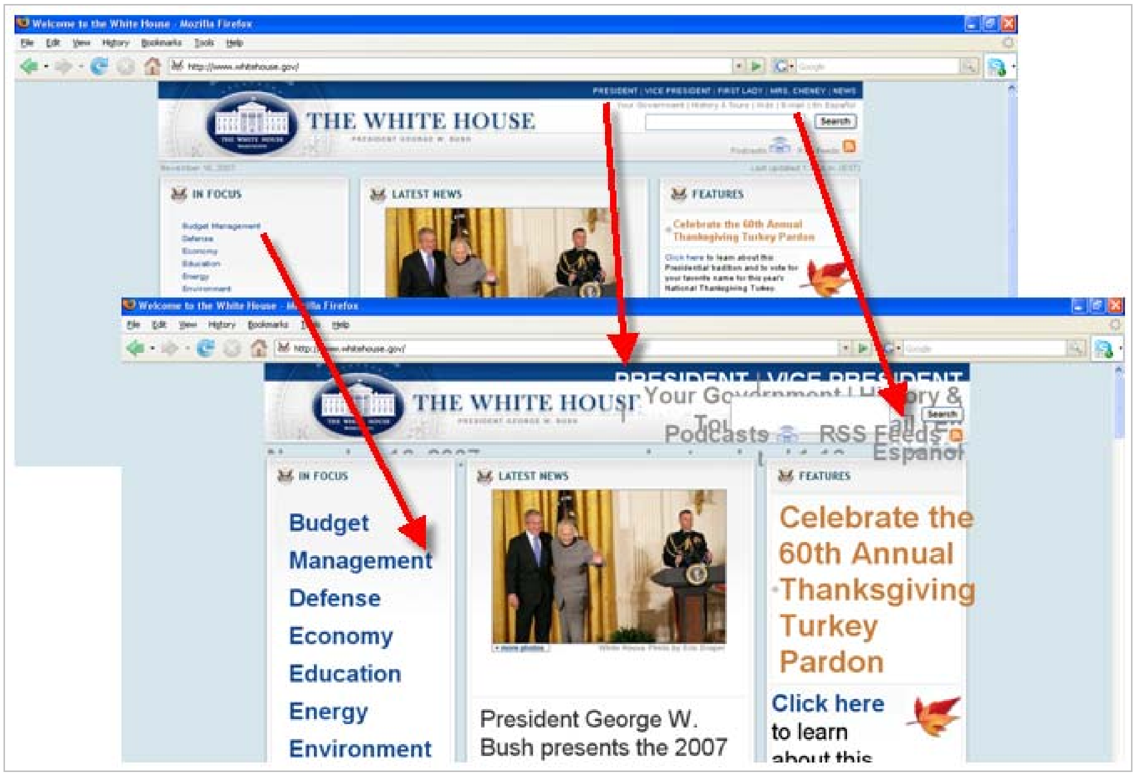


Figure 8: Containment error: increased text size eliminates access to navigational links.

At its default text size, the horizontal bar at the top of the page offers access to "PRESIDENT," "VICE PRESIDENT," "FIRST LADY," "MRS. CHENEY," and "NEWS." With the text size increased four times, only "PRESIDENT" and "VICE PRESIDENT" are partially visible; with enlarged text crowding the fixed-size Web page, the other top-row labels cannot be displayed. The second-row hyperlink labels are also garbled and obscured, with "E-mail" visible only as "ail". The left-side navigation bar, surrounded by unused white space when the text is displayed at its default size, is only minimally disrupted by increasing the text size, with "Budget Management" appearing to become two items rather than one; shorter labels in this area of the page are actually able to support the spirit of the human reader's request to improve readability by increasing text size.

For Web publications, the likelihood of containment errors of this kind is high because there is no central control over the mediating technology that enables reading of Web pages. For a text published as a book, page size is fixed by the publisher; the reader might choose to read a book in any of several different page sizes—hardback, pocket paperback, or large-print edition—but the number of options is small and finite, and the publisher has custom-fitted the contents (text) to each of its containers (pages in a book). For the same text published as a Web page, variation in the attributes of potential containers is effectively infinite, as readers replace publisher-designed paper with screens of their own choosing. Web pages are displayed on screens with physical attributes that are both fixed by the manufacturer—maximum size and resolution—and varied by the user—window size, text size, brightness—as well as software-enabled customizations which mean that two different readers may never read the "same" page. Web-based aggregation services such as *Google News* must handle an even greater level of variability than that created by moving a single publisher's printed text to a Web page. As discussed in the following case study, by collecting news stories from a large number of independent publishers and attempting to automate their placement into a consistently-shaped (but customizable) container, *Google News* provides opportunities to explore some of the errors caused by mismatches between contents and container.

Case Study: Nature of Contents Is Incompatible with Nature of Container

At news.google.com, stories are collected from "4,500 news sources updated continuously"; the basic layout of the news.google.com home page is shown, at a greatly reduced size, in Figure 9. Adapting the structure of a printed newspaper, which may physically separate major sections such as "Business" and "Sports," Google News presents a containerized home page subdivided into sections such as "Top Stories," "World," "Business," "Elections," "Sci/Tech," "Sports," "Entertainment," and "Health." Readers are also offered forty-two "International Versions," including "Estados Unidos" which presents Spanish-language news for a US audience. Each section contains a frequently-changing selection of stories, summarizing three stories in each section, with hyperlinks providing access to the full text of that story and related stories.

In addition to choosing an "international version" of the news, visitors are invited to customize the presentation of news so that the order in which the sections appear reflects individual priorities and interests. If a visitor is most interested in sports and least interested in business and allows Google to record that preference and recognize the visitor's identity, stories can be presented in the preferred order every time the visitor returns to the page, to the extent that uninteresting categories can be completely hidden and highly-interesting categories can be expanded to a user-selected size. Two simultaneous readers of the "same" Web page, news.google.com, are unlikely to be offered the same text.

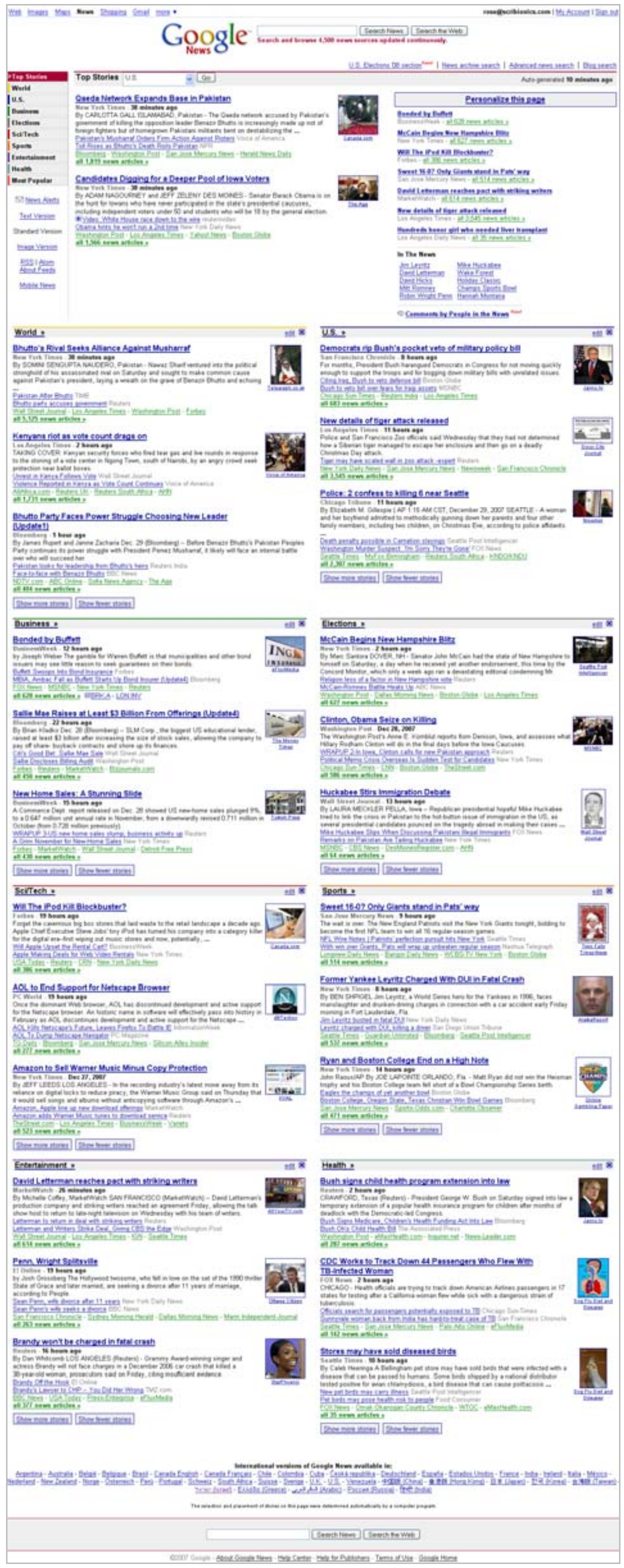


Figure 9: Basic layout of Google News, showing multiple layers of containerization.

True to Google's origin as a search engine, news.google.com can perform keyword searches in a subset of the Web limited to news sources; the "Help for Publishers" link at the bottom of the page explains how a news source is distinguished from other Web sites and provides suggestions to improve the likelihood that a news story will be recognized as such and included in *Google News*. Each story included on the home page is presented in the same format:

- a headline, hyperlinked to the full story at the news source's Web site;
- the name of the news source (often a site associated with print or broadcast news media such as a newspaper or television network; Web-only news sources are also included);
- a timestamp indicating how recently the story has been updated;
- the author's byline;
- the first few lines of the story's lead paragraph;
- a topically-related thumbnail image if one is available;
- hyperlinks to other versions of the story as published by other news sources.

All this is done via software, as announced at the bottom of the *Google News* home page: "The selection and placement of stories on this page were determined automatically by a computer program."

Because no humans are involved in decisions about which sources and which stories to include, Google can disavow responsibility when biased or unpopular or inappropriate viewpoints—"news" stories provided by purveyors of hate or fraud—are included as valid news sources alongside the work of respected journalists. Humans are prone to error, but human error is of a different nature than computer-generated error; automated creation of a news page enables automated presentation of error, as shown in Figure 10 and Figure 11.

Figure 10 is an example from news.google.com in which a story's container is correctly shaped but its contents are invalid. The standard elements of the story as required by the *Google News* format are in place, with one exception: rather then the first few words of the story, a warning about inappropriate software ("We're sorry, your Flash browser plug-in appears to be out of date") appears as a summary of John McClain's story.

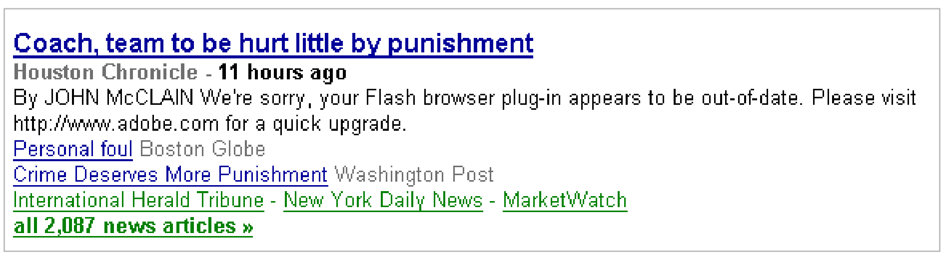


Figure 10: Containment error: software error message in location intended for story text.

When the same visitor using the same "out of date" software clicks the hyperlinked headline to see the story as presented by the news source, though, the *Houston Chronicle* provides the appropriate news story, with no difficulty and no warning message. The software on the visitor's computer is evidently not really out of date, but the *Google News* software misidentified that conditional warning message from the *Houston Chronicle*'s site; rather than retrieving the text of the news story and placing a portion of it into the container reserved for that purpose at news.google.com, Google's software retrieved the text of the warning and presented it in the story's place.

Placement of the wrong text into a story's container is unusual—most of the time, *Google News* presents news stories rather than warnings about software—but it is far from a unique event. As shown in Figure 11, text describing software problems is not the only non-news text that can be inappropriately placed in the space reserved for a news story.

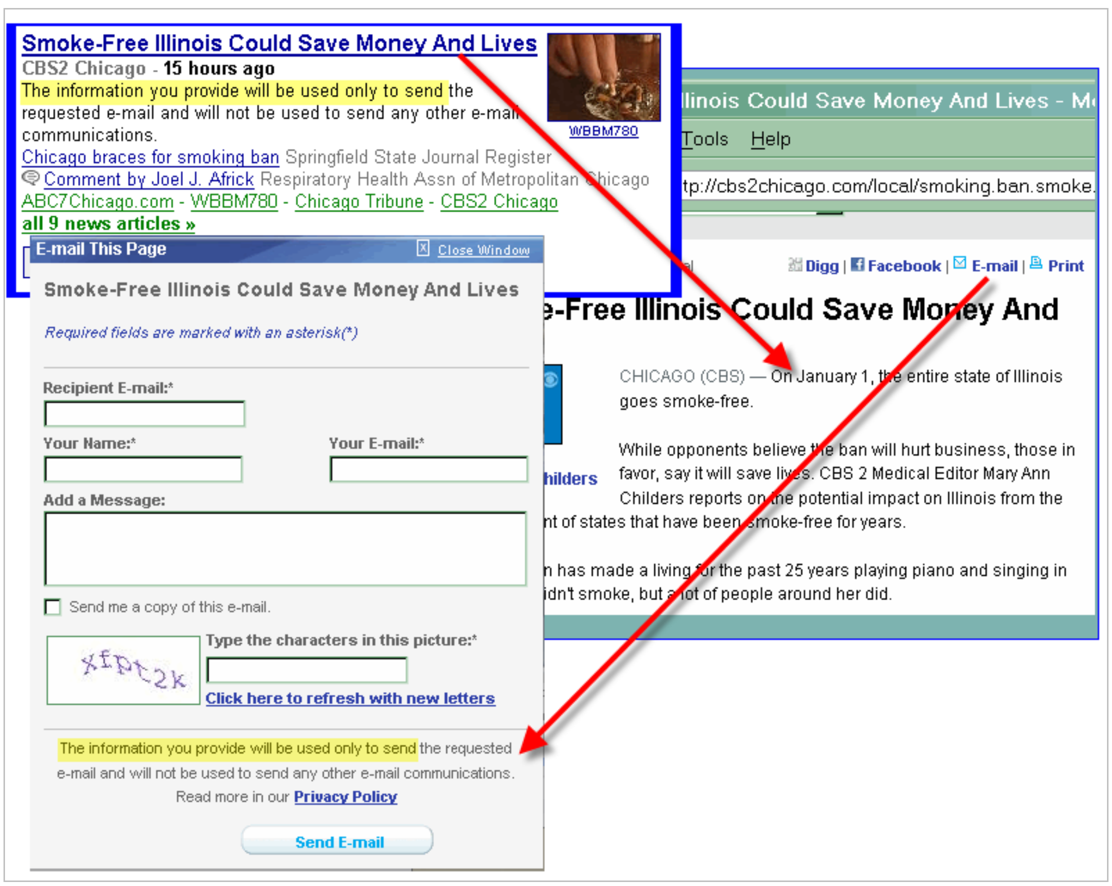


Figure 11: Containment error: guidance to email sender in location intended for story text.

As this example shows, non-news text that could be a legitimate response to a visitor's activity (in this case, a request to send a copy of the story by electronic mail) can be treated by *Google News* as if it were the news story. Here, although *Google News* presents text apparently unrelated to the subject matter of the headline, clicking the hyperlinked headline opens the relevant *CBS2 Chicago* news story; that story, placed within *CBS2 Chicago*'s standard format, offers several hyperlinks to optional tools, including one marked "E-mail"; clicking the "E-mail" hyperlink opens a window containing several items of non-news text, including the text that *Google News* erroneously included as the story's lead paragraph: "The information you provide will be used only to send the requested e-mail and will not be used to send any other e-mail communications."

Whether such errors are due to shortcomings at *Google News* (their software encounters unexpected circumstances and responds inappropriately) or inconsistencies at news sources such as *Houston Chronicle* and *CBS2 Chicago* (they store non-news text in locations where *Google News* expects to find only news), they demonstrate the relative ease with which containment errors can occur, even within a heavily-used Web service from a robust and successful provider such as Google. These are not bizarre, unique errors in experimental pages created by students or novice programmers, but represent patterns of error in a leading search engine's incompatibility with the Web publications of professional news media. The Web has no central management, no Editor-in-Chief enforcing a house style, so there is nothing intrinsically invalid about any news site's choice to structure its Web pages in one way or another; inconsistency among news sites only becomes problematic when an aggregator such as *Google News* attempts to package differently-structured contents into similarly-structured containers.

In both examples from *Google News* discussed above, I believe this is the common factor: Web sites publish digital hypertexts of the news stories that they might also distribute in print or on television broadcasts, but they add non-news tools and features which assist human visitors but which confuse software. The questions raised by Ringlstetter et al. as to pervasiveness of error in human language on the Web, and therefore the usefulness of the Web as a corpus for natural-language research, deserve to be asked in multiple new directions: for example, to what extent are computer-generated texts (such as those created by *Google News*) capable of correctly interacting with humans or computers, and what errors are created by placing the computer in the position of an active author, generating new texts rather than passively displaying existing texts?

It is also possible, as shown in two views of the *New York Daily News* "Barack Obama Keeps Fighting" story in Figure 12, that placement of inappropriate contents into a container is not due to error on the part of those who created the software by which contents and container are matched, nor due to error on the part of those who format and label contents so as to enable appropriate matching. What appears to be a mistake can actually be a choice, what practitioners call "gaming the system": using the rules by which a system operates in order to cause the system to produce a result which its designers did not intend.

Figure 12: Containment error: caption in location intended for story text.

In the *Google News* Elections section for January 10, 2008, something unusual has happened to the *New York Daily News* story. The two other news stories in the section—from *CNN* and the *New York Times*—operate in the standard way, preceding the story with a byline ("NEW YORK (CNN)" or "MICHAEL POWELL MANCHESTER, NH") and then three lines (what *Google News* calls a "snippet") of the first paragraph, and accompanying that text with a relevant illustration. Google independently selects a story and an illustration to feature, so they are unlikely to come from the same source; in this case, the *CNN* story is accompanied by a photograph credited to *Turkish Press*, while the *New York Times* photo is paired with a photo from the *Albany Times Union*. However, as on many other occasions, the *New York Daily News* story is introduced in *Google News* alongside the photograph that accompanies it at its source, the *New York Daily News* Web site.

One clue as to how the *New York Daily News* creates this unusually-strong pairing of its words and images is evident when the story is opened: the text that appears in *Google News* is not the text of the news story, but the text that captions the story's photograph. As perceived by the *Google News* software, the caption *is* the story here. Under "The snippet you display for one of my articles is incorrect," Google's "News (publishers) Help" at www.google.com/support/news_pub/bin/answer.py?answer=70752&topic=11674 explains why this occurs:

> Our automated system looks at each article's source code to extract an appropriate snippet of text to display in *Google News*. If our crawler finds many large pieces of text in the source code of your article pages, it may "guess" at an appropriate snippet.

The twenty-six-word caption under the photo of a young Obama supporter qualifies as a large piece of text, which is enough to misdirect Google's "guess" as to which of the several blocks of text on the story's Web page is actually the story, and to tie the photo strongly to the story and its headline so that they all appear together on the *Google News* home page. Google's software is not open-source, so it is not possible to examine its internal operations in detail to identify the exact lines of code which cause this to occur, but it is possible to observe the pattern of results: lengthy captions are consistently favored above the actual text of a story as *Google News* chooses what to include on its home page. If the confusion is intentional, with the original publisher deliberately manipulating the re-publisher to generate a text that follows its programming rules (large blocks of text are favored in the selection of stories) while violating its design intentions (featured images are selected independently of featured stories), I call the result a *containment gaming error*.

Claims of Internal Unreliability: This Made Sense to a Computer

The *Google News* home page at news.google.com begins with an announcement that its content was “Auto-generated” and concludes with a disclaimer: “The selection and placement of stories on this page were determined automatically by a computer program.” By emphasizing the absence of human agency here (“by a computer program” being only a slight narrowing of “automatically,” which is a repetition of “auto-generated”), Google shapes reader expectations: the information provided here may not be what a human reader would expect to find if that information were managed by humans. As the examples above show, considering *Google News* as a container which may or may not deliver appropriate contents provides insight into several varieties of error with which Web pages, especially those which collect and re-publish material from other sources, may demonstrate.

Example: Container Fails to Identify Nature of Contents

Containment errors can occur when, by intention or by accident, Web pages are designed so that the nature of their content is not clear and that content is extracted and re-published, as it can be with *Google News*, in a way that the re-publisher may not have intended. Error of this nature is not limited to hypertexts; when any work moves from one container to another, it is especially vulnerable to this kind of confusion. Travel is the pervasive metaphor of the Web, but not all text travels well.

To take a famous example, readers of the printed text of H.G. Wells' 1898 novel *The War of the Worlds* could not have interpreted it as anything but a work of fiction: an exciting story, but clearly an imaginative rather than an informative text, nothing that required any reaction. When the story was formatted as a series of newscasts and presented as a Mercury Theater radio drama in 1938, many listeners misinterpreted the performance as actual news reporting (information) rather than as a dramatic performance of news reporting (imagination). The potential for such confusion is caused by the different natures of the printed novel and the radio drama as textual containers. A printed book is a strong and inescapable container, with pages firmly bound together and no possibility, even if a reader opened the book at random and began reading the middle of the story, of interpreting a page independently of the labels applied to the book's cover; a page of a novel might be mistaken for a page of a similarly-shaped memoir, but it cannot be mistaken for a page of a newspaper. A radio broadcast, though, is a weak and leaky container: if a listener, having missed the context provided by the broadcast's spoken introduction, tuned in mid-performance, there would be no clue as to the nature of the container (a play, not a report) and therefore the nature of the contents (imaginative, not informative) and therefore the nature of an appropriate response (have some hot tea and go to bed, not grab a shotgun and hide in the basement). Absence of an obvious container prevents establishment of audience assumptions: unlabeled and decontextualized, textual contents are often insufficient to establish the text's meaning.

When a printed news story is moved from a newspaper page to a Web page, it moves from a strong container (a newspaper, not mistakable for anything else) to a weak one, becoming vulnerable to the same kinds of confusion that attached to *The War of the Worlds* when the printed novel re-appeared as a radio drama. When a printed text becomes a Web page, it acquires the attributes of Web pages, including an ease of re-publishing that does not exist for paper publications. Another example from *Google News*, shown in Figure 13, illustrates a kind of error that I call *containment association error*.

Google News' January 11, 2008 Elections section offered a *New York Times* news story written by Carl Hulse about a Congressman, "[t]he highest-ranking African-American in Congress", accompanied by a seemingly-bizarre photograph of chickens. The *New York Times* story included a large, accurately-captioned photograph of the Congressman; other current news stories in the same *Google News* section offered photographs of subjects closely related to the text of the story (civil rights activists, presidential primary candidates, African Americans, Congress, South Carolina); with so many illustrations available from Web-based news sources to accompany this story, why was the story about the Congressman paired with a photograph of chickens?

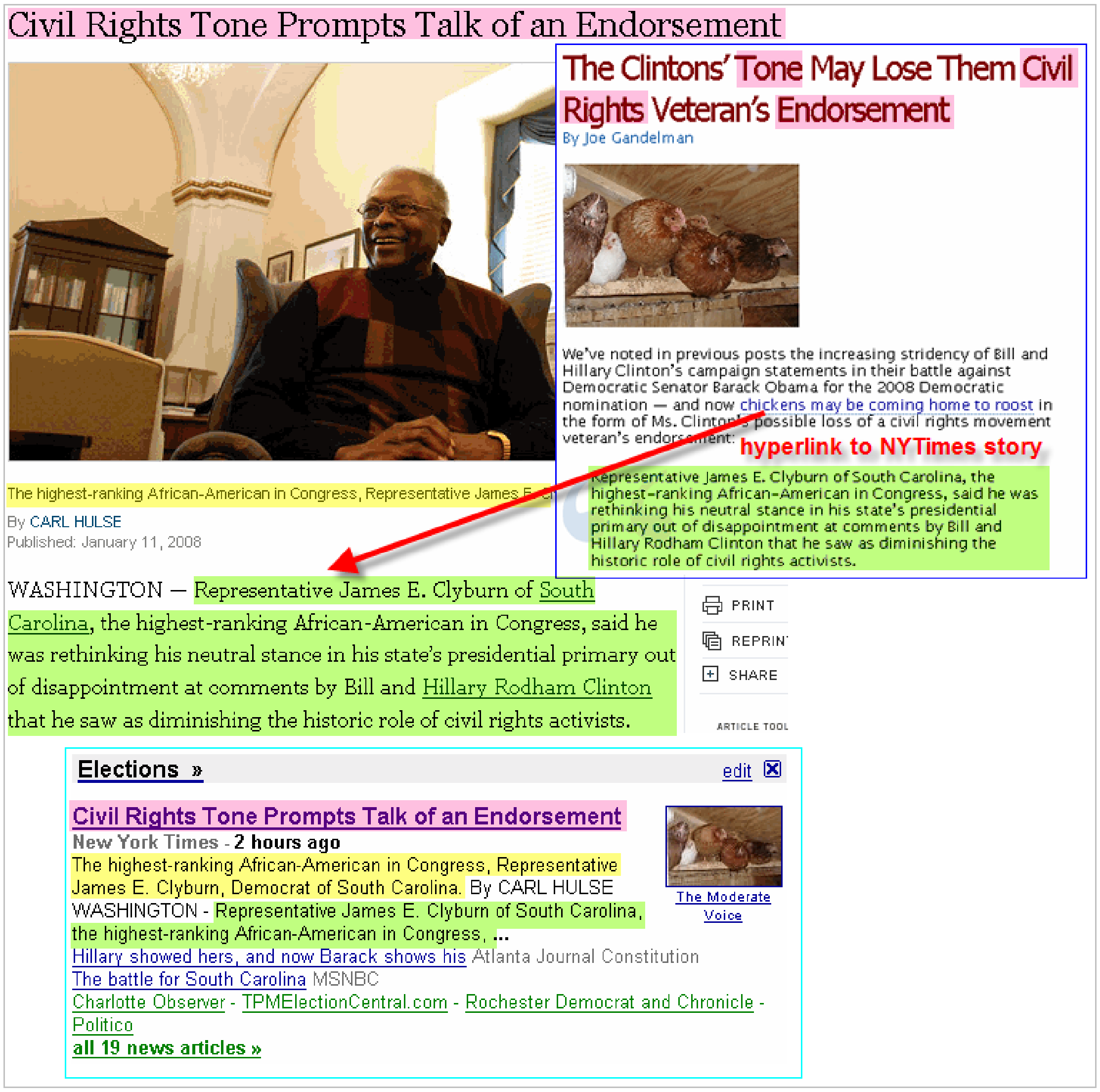


Figure 13: Containment error: serious text inappropriately combined with humorous image.

The cause of the strong connection between the story about the Congressman and the photograph of chickens can't be proven without examination of Google's proprietary software, but it can be hypothesized after close examination of the photograph's source, a blog named *The Moderate Voice*. The blog comments on the *New York Times* story and hyperlinks to it from its own statement that "chickens may be coming home to roost," a folk expression meaning that past mistakes will have to be dealt with; a photograph of roosting chickens accompanies that comment, as do a few lines of text reproduced (without attribution) from the *New York Times* story. For *The Moderate Voice*, *Google News*' high-visibility presentation of the connection between their words-and-image about chickens (representing mistakes) and politicians (not the Congressman, but the candidate whom the Congressman may no longer support) may be a sign of strength, showing that their blogger (Gandelman) can compete successfully against traditional journalists for readers. For *Google News*, though, the same pairing of serious story and humorous photo may be a sign of weakness, demonstrating that, unlike news sources managed by human editors, *Google News* cannot appropriately recognize and categorize such human linguistic and cultural complexities as humor, aphorism, satire, and metaphor, and can in fact be gamed (thanks to widely-circulated information about techniques, such as hyperlinking and illustrations, that cause Google to rank one page more highly than another) into creating strange juxtapositions of textual and visual elements that it incorrectly perceives to be natural pairs.

Keyword-based automated selection of advertising text to accompany a news story operates on the basis of a similar assumption about natural pairings of texts: if a reader is interested in the subject matter of a news story, and an advertisement relates to similar subject matter, the reader may also respond to the advertisement. The difficulty here is that, while software can confirm the general status of "interested" by observing that a reader has opened a Web page, it cannot distinguish between subcategories of that status. "Interested and attracted" might indeed yield some follow-up activity, seeking more of the same by clicking on an advertiser's hyperlink; "interested but horrified," though, is more likely to be revolted by the automated commercialization of tragedy, as demonstrated in the *Austin American Statesman*'s January 13, 2009 "Helicopter Crash on A&M Campus Kills 1, Injures At Least 5" story; at the bottom of the Web page, "Ads by Yahoo!" offers helicopter flying lessons, helicopter tours, and toy helicopters. This is another association error, an inappropriate pairing based on words ("helicopter crash" and "helicopter lessons") that relate to the same idea but do not appeal to the same reader. At a surface level, ideas can give every indication of being strongly connected based on expression in similar words; humans, though, have access to a deeper non-digital reality in which words are often weak and misleading signals.

Claims of Impermanence and Limitation

Some designers and developers of Web pages address the instability of their texts by clearly labeling them as unstable, notifying the reader that the text is unfinished or otherwise unreliable. Setting expectations in this way may prevent the text from ever being identified as erroneous: if it is designed to be unreliable, and it is unreliable, then it meets its design requirements and is operating correctly.

Example: This May Be True, for Now, but Don't Quote Me

Made-for-the-Web news stories, and video and transcripts of made-for-television programs originating on the *Fox News Channel*, are published at www.foxnews.com. Digital video can be copied to the Web directly and reliably, but creation of a transcript requires multiple time-consuming and error-prone steps—typing, editing, posting on the Web—so transcripts on the Fox site are bracketed between expectation-lowering statements. At www.foxnews.com/story/0,2933,483415,00.html, the transcript of an interview by Greta Van Susteren entitled “Gov. Blagojevich Goes 'On the Record': 'The Fix Is In'” is introduced with a disclaimer (“This is a rush transcript from "On the Record ," January 26, 2009. This copy may not be in its final form and may be updated.”) and concludes with another disclaimer (“This is not a legal transcript for purposes of litigation.”). Foxnews.com makes its position on this unstable text very clear: the transcript is informational but it may not be factual.

Example: If You Leave, You Are on Your Own

A Web page is contained within a Web site, but this containment is weak; Web pages routinely provide their visitors with hyperlinks to external pages which are outside the control of the linked-from site’s webmaster. Creating a hyperlink to a page at an external site is no more technically difficult than hyperlinking to a page within the same site and requires no coordination with the webmaster of the linked-to page. However, hyperlinks that take the reader away from some sites introduce a structural hesitation, an additional step warning of unpredictable results at the external site. For example, clicking on a hyperlink from a site operated by one agency of the U.S. government to a site operated by another agency of the same government does not result in a direct transfer to that site. Instead, a page such as Figure 14 is displayed.

Figure 14: Web page warns that going elsewhere creates risk.

In this case, although the Internal Revenue Service site at www.irs.gov offers a direct path to the Social Security Administration site at www.ssa.gov, the usual hyperlinked structure is deformed; the requested Social Security page is displayed only after the additional step of clicking the “Leave IRS Site” button. Warnings of this nature allow the page’s publisher to operate as a full-service information provider by providing hyperlinks to sources of information beyond its internal limits, while also disclaiming responsibility for any ill that befalls a visitor who chooses to follow one of the offered hyperlinks; the purpose is not to hold the visitor within the confines of the Web site, but to contain the scope of responsibility.

Claims of Loss: This May Not Be the Best Copy

Films designed for presentation on a large, public theater screen but displayed on a small, private television screen routinely begin with a warning to viewers that this is likely to be an unsatisfactory experience: "This film has been modified from its original version. It has been edited for content, to fit on your screen, and to run in the time allowed." Similarly, digital representations of texts that originated in other media often provide introductory expectation-lowering warnings such as those in Table 2.

Table 2: Some digital representations of non-digital texts warn users not to expect perfection.

Source	**Disclaimer**
Association for Computing Machinery Digital Library	OCR errors may be found in this Reference List extracted from the full text article. ACM has opted to expose the complete List rather than only correct and linked references.
University Microfilms (.PDFs of microfilms of paper)	THIS DISSERTATION HAS BEEN MICROFILMED EXACTLY AS RECEIVED. *or* Some Pages have indistinct print. Filmed as received. *or* The quality of this reproduction is dependent on the quality of the copy submitted.
ProQuest Historical Newspapers	Some Adobe Reader functionality is not available for Scanned Image PDFs. This includes: Copying and pasting text; Searching within the text; Reading text using screen reader software.
NetLibrary eBook	Some images in the original version of this book are not available for inclusion in this NetLibrary eBook.

A digital representation of an imperfect original is itself perfect because it does not disrupt the imperfections of the original; if the text was blurred on paper, it will be blurred after scanning to electronic paper. In the case of eBooks, a digital copy may be imperfect for legal rather than technical reasons. For example, the printed version of Jeffrey Berman's *Narcissism and the Novel* features a large image on page ii, captioned "Narcissus by Irving Amen. By permission of the artist"; as an eBook, the image is replaced by additional text (“Image not available.”), after which the original caption crediting the artist is faithfully reproduced so that the eBook deviates as little as possible from the printed book. On one level, page ii becomes nonsensical (“Image not available.[...]By permission of the artist.”); on another level, these combined statements clearly define what was lost when the printed book was transferred into a digital container.

Claims of Passivity: “Powered by”

Web pages are often constructed by software; humans create page-generating software to work for the general case, then other humans use the software to create Web pages for their own purposes. Working within (“powered by”) a system that generates dynamic pages, the human implementer’s role is to establish general site-wide assumptions rather than to arrange specific correct handling of difficult-but-infrequent data. The difference between a software-generated page and a hand-coded page is the difference between an automatic security-camera photograph and a posed portrait in a photographer’s studio. That difference is not accuracy (both photos capture a real scene, and either can be retouched) and is not planning (in both cases, the photographic situation is arranged to serve a specific human purpose); the difference is active human agency, the interfering hand of the photographer. Similarly, pages created by general-purpose software may not be perfectly adapted to readers and situations that deviate from the general case; where a human author did not make adjustments to correctly publish an unusual text, a human reader may have to make mental adjustments to interpret the resulting noisy text.

Case Study: Coding Errors in TAMU Faculty Personal Pages

Software-generated Web pages can be imperfectly formed in non-standard situations such as the use of non-English characters in English-language database records, as shown in Figure 15.

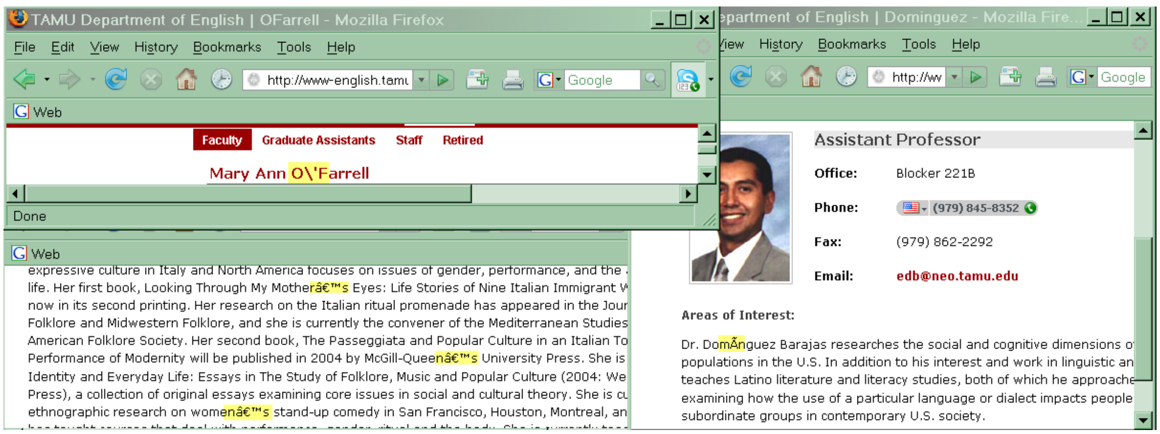


Figure 15: Web pages showing errors in representation of special characters.

These three examples of faculty pages from www-english.tamu.edu, the Department of English at Texas A&M University, demonstrate the ease with which non-English and non-alphabetic symbols can complicate the accurate representation of a digital text. Mary Ann O’Farrell’s page at incorrectly presents her name as “O\’Farrell”, and Elias Domínguez Barajas’ page garbles his name as “DomÃnguez”. Giovanna Del Negro’s page includes errors related not to her name but to punctuation throughout the page: all possessives are mis-handled, so “Mother’s,” “Queen’s,” and “women’s” now all end in “â€™s” rather than “’s.” All these pages are marked visibly with the name of the software used to create them (“Powered by MODx”) and named in a way (index.php?id=163, for example) that identifies them as dynamic pages: none of these Web pages exist as a static whole but each is assembled upon request from a database containing general information about each faculty member and placed into a container for human-readable display. Information describing Dr. Domínguez Barajas is requested by sending a request (“id=163”) to the software; the container in which the requested information is displayed is a Web page named “index.php.”

Errors such as shown above in Figure 15, in addition to indicating the extent to which general-purpose software rather than a human designer controls a digital text, also suggest that a text originated in some digital container other than the one in which it is currently presented. HTML includes methods of encoding non-English and non-alphabetic symbols. For example, HTML Unicode would represent “Domínguez” as “Domínguez” or “Domínguez”; absence of this encoding indicates that the text that did not originate as HTML. In PHP, the language in which MODx is written, paired inverted commas are used to separate computer-directed code from human-directed messages, so that a PHP statement such as echo ‘hello’ displays the message “hello”; it is possible in PHP to properly handle non-paired inverted commas and display messages such as “O’Farrell” and “women’s” but the encoding required for this, as with HTML Unicodes, is not automatically created when text from an external source such as a word-processing document is copied into a database. Encoding errors of both types are signs that a Web page contains text created for a previous container and not adapted to suit its current container.

Reliable Texts, Valid Software, Contextual Errors

The MLA's Committee on Scholarly Editions, in its *Guidelines for Editors of Scholarly Editions* at www.mla/cse_guidelines, points out that the "[t]he scholarly edition's basic task is to present a reliable text: scholarly editions make clear what they promise and keep their promises"; a text's *reliability* is established by the criteria of accuracy, adequacy, appropriateness, consistency, and explicitness. For software, a similar key value is expressed as *validity*: software is valid if it meets the requirements established for it. Software requirements, written in human language, explicitly state what the software must accomplish ("make clear what they promise"); software validation compares the actual operation of the software to those requirements, confirming that the software functions correctly ("keep their promises"). For software engineers, software is *valid*, in that it meets its requirements, even if it is also susceptible to errors unrelated to those requirements; if software works correctly when it does work, but sometimes fails to work at all, it is valid but unreliable. Software becomes *reliable* as its errors are identified and corrected, with reliability narrowly quantified as "[t]he probability of failure-free operation of a computer program for a specified period of time operating in a specified environment" (Rakitin 168). Increasing *reliability* does not increase the software's *validity*, but it can increase its *usability*, the degree to which it is accepted and put to work by its users.

Because digital hypertexts are texts mediated by software, their errors must be considered according to the criteria separately relevant to texts (*reliability* established by accuracy of content, as in accurately transmitting an author's words) and software (*validity* established by meeting functional requirements, such as correctly processing one user's request; *reliability* measured by success over time and in context, such as correctly processing ten thousand users' requests). Considering hypertextual errors in this multi-pronged way forces recognition of hypertexts' dependencies on unstable external factors: even if the information to be presented in a hypertext is correct (*reliable*, as that term is applied to texts) and the mediating software is correctly designed (*valid*) so that it can convey the appropriate information in response to a user's request, some or all of the information may never reach its intended readers. Figure 16 illustrates that a hypertext, in this case an HTML-format newsletter e-mailed from www.poems.com to subscribers, can be partially defective in ways that vary depending upon context (*unreliable*, as that term is applied to software), even when its verbal elements are correctly conveyed.

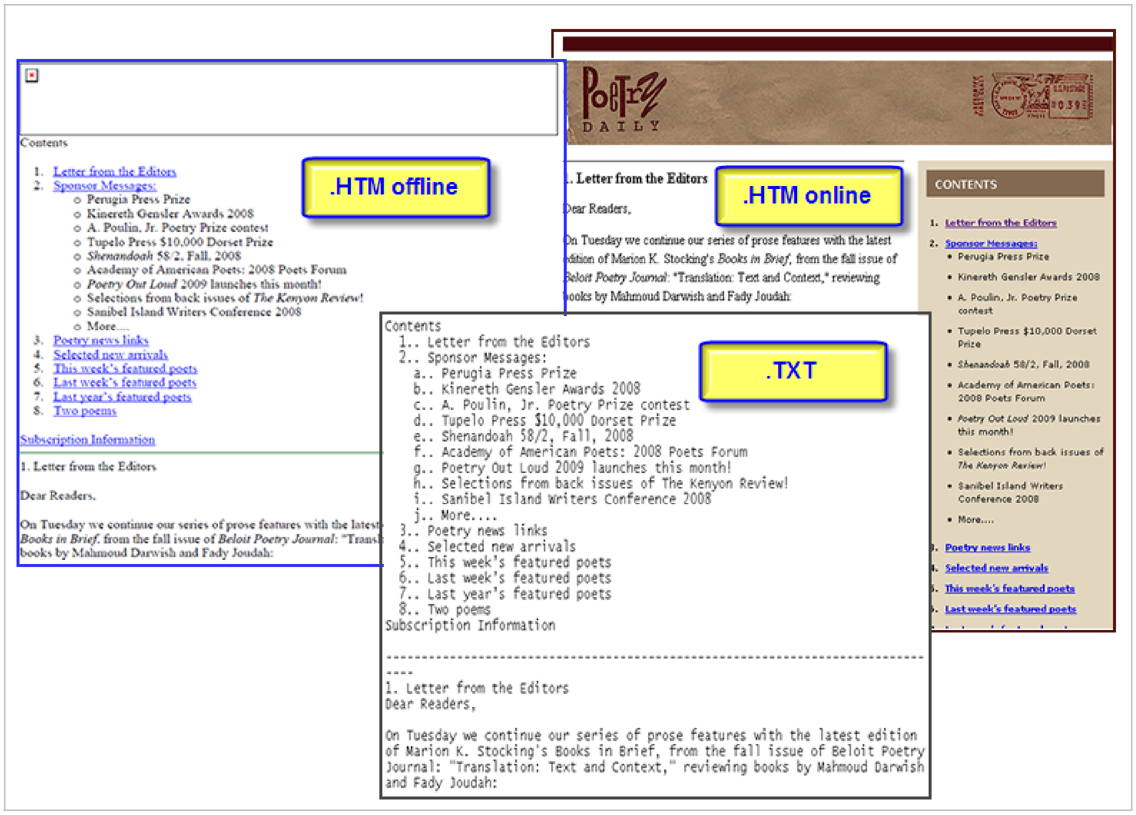


Figure 16: Hypertext varies when viewed offline (L), online (R), or as plain text.

This figure presents the same digital newsletter formatted in three ways, with the same words conveyed in every format; however, only the rightmost example, marked ".HTM online," is completely successful in conveying its author's message if, as I argue, that message properly includes matters of structure and format. The ".HTM online" example presents "Letter from the Editors" as the first story in the newsletter, with a "CONTENTS" section in a colored box beside it; the colors of the box, matching the color scheme of the "Poetry Daily" logo and other artwork in the newsletter but differing from the black-on-white of the newsletter's stories, visually categorizes "CONTENTS" as supporting material, useful in understanding and navigating the text but not meant to be read as a story. The newsletter's HTML source code shows that its appearance is controlled by an external stylesheet at www.poems.com/css/pd_news.css; if that stylesheet is not available, perhaps because the reader's personal computer or the poems.com server is not connected to the Web, the formatting instructions are not available and the ".HTM offline" appearance is the result.

While the structural defects created by the stylesheet's absence may be unnoticed by a reader of the newsletter, the "HTM offline" example is partially dysfunctional on a more visually obvious level: hyperlinks to locations within the newsletter operate correctly, but hyperlinks fail to include externally-stored decorative images such as www.poems.com/images/mailout/header.jpg; space reserved for the image contains diagnostic information instead, with a boxed red "X" indicating that an image specified in the design is unavailable. This occurs because, as e-mailed to its recipients, the newsletter contains only its verbal elements and the ultrastructural elements necessary to point to design elements such as images and stylesheets which are maintained externally. This is a widespread practice, reducing the size of e-mail messages, but it is complicated by another widespread practice: after downloading e-mail messages, a reader can choose to disconnect from the Web and read the messages while offline, severing the intended connection between elements internal to the HTML-formatted e-mail and elements external to it.

For HTML-formatted e-mail such as the *Poetry Daily* newsletter, splitting the necessarily-online activity of receiving the text from the possibly-offline activity of reading it creates an unnatural separation between its self-contained words and its externally-controlled structure, causing the text to become defective when read offline. The ".TXT" example at the center of Figure 16, acquires additional defects by being read as plain text rather than hypertext. For improved efficiency and reduced risk of infection by viruses, electronic mail software (such as Microsoft Office Outlook) commonly offers readers the option to convert HTML-formatted e-mail to plain text; viewed as plain text, the hyperlinks to locations within the newsletter no longer exist, and the absence of an external image is no longer reported, increasing the newsletter's distance from what its author created and decreasing the possibility that the reader may notice that something is missing.

The ".HTM online," ".HTM offline," and ".TXT" examples contain identical words, differing only in design and navigation elements; I argue that these differences, whether created by environmental accidents or a reader's conscious decision, are errors because they create for the reader a version of the newsletter which substantially differs from what the newsletter's author created. When caused by an accident such as a network outage, these are *connectivity errors*; when caused by the reader's request to override the author's design, eliminating structural, navigational, and decorative content, I describe the result as an *exclusion error*. I argue that a text with partially-missing contents is defective, even when the defect was created at a reader's request; to cause an exclusion error, the reader must consciously choose to delete content rather than, as discussed earlier in this chapter, accidentally losing content by changing font size or otherwise pushing it out of its container. Exclusion errors can be transitory, easily corrected by connection to the Web or selection of a different software option; they can also easily become permanent in copies saved as plain text, captured as screen images, or printed on paper or electronic paper. The complications introduced by copies, especially when textual contents are preserved but the structures containing them are altered or lost, are the focus of Chapter VI.

Managing Errors in Hypertexts: Proofreaders, Testers, Engineers, Editors

All methods of textual production can create texts that contain errors. For printed texts, pre-publication prevention and post-publication correction of errors are the basis of several professions: copyeditors, fact checkers, and proofreaders examine pre-publication drafts to prevent errors from appearing in the final, published text; scholarly editors create new, reliable texts to improve upon older, defective texts. For software, too, prevention and correction of errors occupies a variety of professionals, including testing technicians, quality assurance analysts, and integration engineers.

Because digital hypertexts are at least mediated (and sometimes created) by software, the methods by which software is validated and maintained may be at least as relevant as the methods by which printed texts are checked and corrected. In particular, I argue that error-detection methods being developed for open-source software are likely to be extensible to Web-based hypertexts, since both are strongly characterized by high-speed, multi-authored change without central control. Like digital hypertexts, personal copies of open-source software can be readily acquired and modified so that each user has a unique copy; at the same time, customized copies (what software developers call "forks") can be offered to the general community in parallel with the general-purpose original, creating the possibility of simultaneous different, but equally valid, versions. In a 2007 study of software error in the open-source Mozilla browser, Koru et al. observed that error-proneness increases as class size increases. If their observations can be extended beyond software to software-created hypertexts, they may mean that, as shown in many of the preceding examples in this chapter, automated data aggregators such as *Google News* are unavoidably prone to error: bundling large numbers of Web pages into a single container ("class"), the more successfully the aggregator collects Web pages, the more defective its collection becomes. While transmission of single digital texts is relatively error-free, aggregation is error-prone. The Web is rich in aggregation services such as search engines, community portals, and syndication streams; these are the contexts in which hypertextual error is most likely to occur, and in which it can most productively be studied.

CHAPTER III

"README"—VERSIONING AND COMPARISON

Error is destructive change, creating a text that is in some way defective as compared to its source in an earlier text or an author's imagination. Constructive change is also possible; for informational texts, constructive change occurs not only in the correction of errors but in expansion to include new facts and updated analyses. For hypertexts, as for their mediating software, constructive change can mean much more than correcting and expanding linguistic content: new behavioral features can be added so that, while the words are unchanged, they become more efficiently searched or more easily commented upon or more usefully connected to related material. This is *hyper*-versioning: creating a new version by constructively changing the ultrastructure that defines the nature and function of a hypertext.

In this chapter, I focus on multi-version informational texts as useful models for tracking change in hypertexts; as I will demonstrate, some methods of identifying and visualizing change are applicable to many kinds of texts, including digital hypertexts and paper books. I argue that, for both paper-like texts such as e-books and software-like texts such as blogs, versions must be compared and described on multiple levels. Hypertextual change must be tracked in ways appropriate to its nature, tracking changed containers separately from changed contents and appropriately identifying the transformations that may occur in the processes of textual production.

Multi-dimensional change can be difficult to fully describe verbally, especially when it relates to changes in structure and sub-surface ultrastructural features; I include and discuss examples of useful visualization techniques as efficient alternatives to unwieldy descriptions and comparisons. Some of these techniques—tag clouds and spider diagrams—are widely used in other fields and are supported by established software tools; the technique of creating bead diagrams to model texts' structural changes is my own development, inspired by physical models of chemical structures.

Digital hypertexts are built with software tools. The rich contextual environment developed around software—itself text written in artificial language, surrounded and supported by other text written in human language—demonstrates the multiple layers of structure and containment required to create an environment that supports flexibility and immediate change. Examining and criticizing software as a text in its own right as well as a container of text is consistent with open-source software developers' view of software as "a new kind of literature that forms part of the common heritage of humanity: to be published, read, studied, and even added to, not chained to desks in inaccessible monastic libraries for a few authorized adepts to handle reverently" (Moody 4). Software versions are customarily identified with multi-level numbers, with the difference between version numbers quantifying the difference between versions; as will be discussed under "Software Is Language Designed to Change," the difference between Version 1.1.1 and Version 2.0 is understood to be profound.

For books, though, versioning is identified holistically, at the highest possible level: the *book* is labeled as the second edition, even if it is identical to the first edition in all respects other than one rewritten *chapter* and a redesigned cover page. Holistic versions are also often identified non-numerically: for example, *The Princeton*

Encyclopedia of Poetry and Poetics, published in 1965, was replaced by an Enlarged Edition in 1974, which was replaced by *The New Princeton Encyclopedia of Poetry and Poetics* in 1993. As “enlarged” relates to size and “new” relates to time, these version descriptions cannot be meaningfully compared; publication date, not the version name, can be used to properly relate multiple versions to each other.

For digital hypertexts, versioning is most appropriately tracked granularly, acknowledging that the multiple layers of the text change independently. Because content-management software is designed to strongly separate online publication into the independent activities of organizing units of content, transforming database-stored content into browser-visualized content, and delivering content to an audience (Addey et al.12), Web pages created by this kind of tool provide a clear demonstration of hypertexts’ multiple levels of versioning. In my own practice, I use the open-source WordPress content-management system, available at www.wordpress.org, to support several weblogs (blogs); examining the multiple layers of one blog will illustrate the necessity of multi-layered descriptions of hypertextual versioning.

Textual Permutations in Blog Publication: Locating Non-Authorial Participants

In *William Shakespeare: A Textual Companion*, Stanley Wells and Gary Taylor illustrate the many versions of a play that were likely to be produced between the author’s invention of the plot and the audience’s experience of a performance; I have re-drawn a portion of their Figure 13, which they title “The permutations of dramatic manuscripts” (31), as the lower portion of my Figure 17.

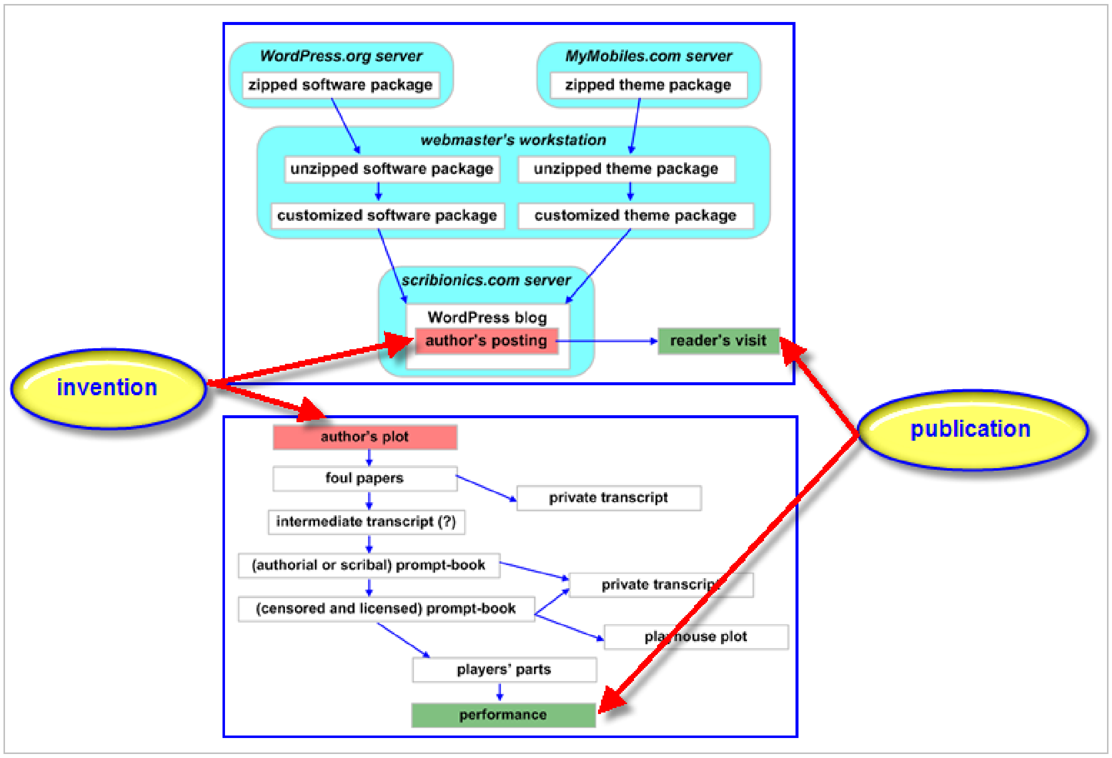


Figure 17: Invention and publication are at different distances from each other for digital blog (top) and manuscript play (bottom).

Following Wells and Taylor's model, I have identified some of the non-authorial steps and textual permutations possible in digital publication by illustrating the construction processes for a WordPress blog which I maintain at www.scribionics.com/blog; that illustration occupies the upper portion of Figure 17.While it is true that "[u]pdating a weblog is fairly easy and can be done quickly without the need for reviewing, editing, or approving" (Bausch et al. 5), it is not true that the author of a blog is independent: "self-publishing" a blog is made possible by the work of hundreds of programmers and designers and at least one webmaster. As Figure

17 shows, digital publication places the author's activity adjacent to the reader's activity only because other participants' prerequisite creative and transformative activities—creating software, designing a container in which the software can present text, downloading and customizing both, and installing them together on the blog's hosting server—are done prior to the author's attempt to publicly communicate an idea. For the play, participants other than the author—scribes, censors, players—operate in the space between the author's invention and the public's experience of the performance. The order of pre-publication events, but not the fact of them, is an essential difference between the Shakespearean play and the modern blog.

For both the digital blog and the manuscript play, a series of steps must be successfully completed to enable the author's ideas to reach the public; at each step, opportunity exists for intentional or accidental change. In both cases, pre-publication processes limit the degree to which the author's invention can be realized. If a blogger wishes to publish a video clip but the blog's software supports only plain text, then the blogger's idea must be revised or abandoned just as if, for Shakespeare's play, the Censor forbade a certain scene or speech. The Web, though, gives the blogger the advantages of decentralization: while Shakespeare could not have applied to a differently-designed Censor for approval, a blogger has multiple possibilities for alternative methods of publishing the video clip, making it possible that the barrier between the author and the public is a permeable one. The blog's content-management software could be modified, adding support for video, or the software could be replaced by a more feature-rich competitor; the video clip could be published elsewhere, at a video-sharing site such as YouTube, and the blog content modified so that, rather than attempting to present the video directly, it provides a hyperlink to the video in its external location.

In a blog, intentional change can occur not only when the author publishes new contents or edits old contents, but when the webmaster installs modified software or activates a different theme. Switching themes can be the equivalent of repackaging any product to increase its attractiveness to consumers, giving its container a new look without changing its contents. However, packaging changes can also be functional, improving the package content's safety (making it child-proof or leak-proof or adding a warning label) or increasing its usability (making it easier for arthritic hands to grasp) or its lifespan (shielding it from the damaging effects of light and heat). For the WordPress blog constructed as shown above, the implications of intentionally changing the theme (container) are discussed and illustrated below.

Case Study: Versioning in a WordPress Blog

Introducing her 2008 *Ultimate Blogs* anthology, Sarah Boxer estimates there are "80-million plus blogs out there" (xxii); clearly, understanding intentional change in this form of hypertext is an important step toward developing a usable theory of hypertextual versioning. A blog is constructed by at least three layers of authorship: authorship of linguistic content, called "posts" or "postings" or "articles"; authorship of the container, called a "theme" or "skin" or "template," in which the content is presented; authorship of the content-management software which creates and modifies dynamic Web pages by combining and managing contents and containers. Figure 18 compares two themes, Pixeled 1.9 on the top and Gear 1.2.4 on the bottom, applied to the same blog.

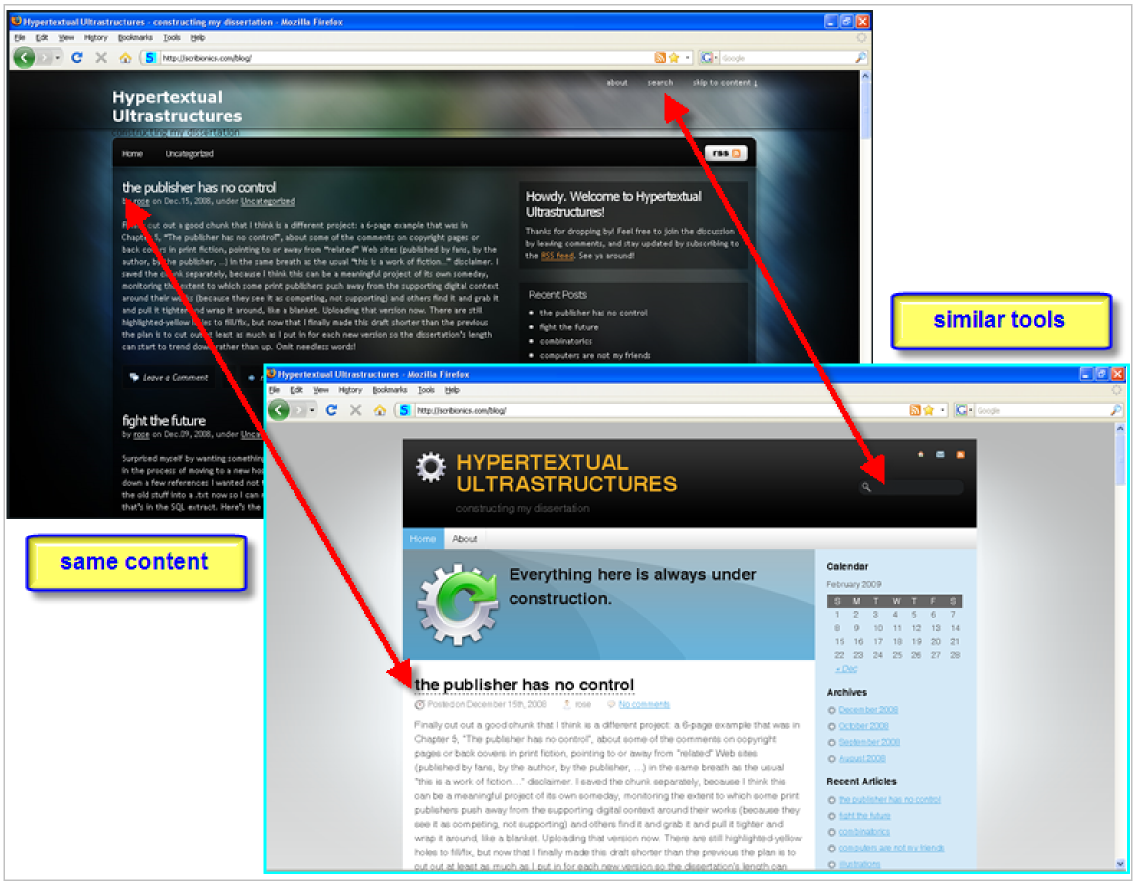


Figure 18: WordPress blog using "Pixeled" (top) and "Gear" (bottom) themes.

To create each part of this illustration, a previously-installed theme was activated for the blog; such a change can be accomplished in less than a minute. While the text of the author-created content is not affected by this transformation of its container, the scope of the change is greater than the obvious differences of color scheme and font size. For example, while both themes provide access to a search tool, the “Gear” theme reserves space in the blog's header, marked by a tiny icon of a magnifying glass, in which to type search terms; the "Pixeled" theme performs the same function by requiring the visitor to click on a "search" hyperlink, which will then provide a new page in which to type search terms. This difference is not one of functionality but of efficiency: searching is possible with both themes, but requires fewer steps with "Gear." Other differences do change the blog's functionality: the "Gear" theme includes a calendar and a mailto hyperlink to the blog's author, while the "Pixeled" theme includes neither.

Themes are designed to be customized, so many of the differences between them could be readily changed by adding and customizing small software components called widgets; this is generally a task assigned to a webmaster (responsible for the container) rather than a blogger (responsible for content.) If this were done extensively, especially if new widgets were created, there is an argument for assigning a fourth layer of authorship, or perhaps editorship, to the blog site's webmaster. Figure 19 illustrates the webmaster's typical activities in choosing, installing, activating, and optionally customizing a WordPress theme.

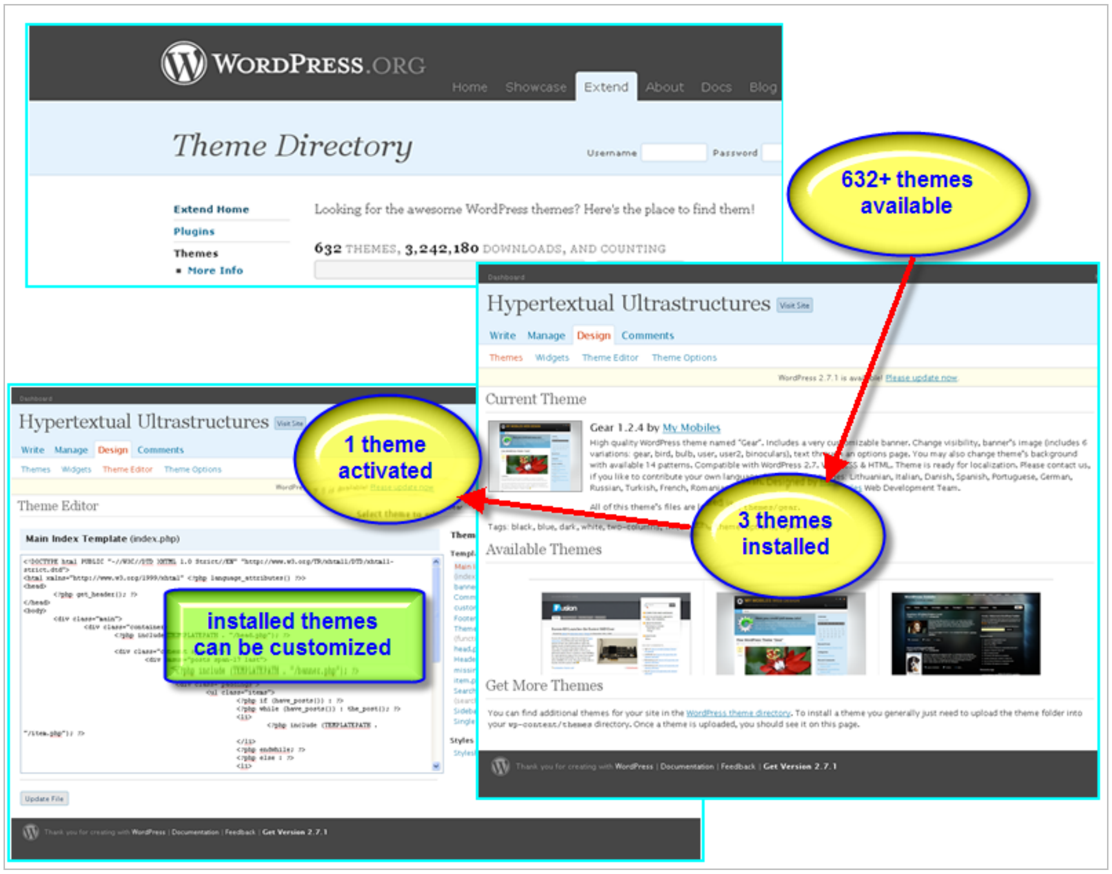


Figure 19: From a public collection of WordPress themes, the webmaster installs several and activates one.

The webmaster did not create WordPress nor its themes but did install and configure them and manages the Web site and database in which they are housed. For the blog that is the subject of this example, since I know that the webmaster’s creative activities were limited to making choices and trivial customizations, I exclude the webmaster from consideration as an author of the blog.

Many would argue for consideration of a fifth layer of authorship, in which the reader of a hypertext, moving from page to page in a sequence that may never have been produced by any other reader, creates a unique combination of textual experiences;

however, for the same reason that I do not consider the installation activities of a webmaster to be those of an author, I argue that a blog visitor's acts of navigation and selection are not acts of creation, and that reading or otherwise using pre-existing content does not constitute authorship.

Access to the Web and thence to the blog is mediated by many additional software tools, including a Web browser (Mozilla Firefox) and the operating systems on the visitor's computer (Microsoft Windows) and the Web site's hosting server (Unix). All these tools are texts, with identifiable authors and versions; differences among readers' experiences can sometimes be tracked to differences in the mediating software—for example, some scripting and markup works differently in Mozilla Firefox than it does in Microsoft Internet Explorer—but, since these are external, environmental differences rather than differences intrinsic to the blog itself, I do not think it is useful to address them as additional authors and additional sources of intentional or accidental change in versions of the blog.

If a blog were a living, speaking organism, its three layers of authorship could be identified as the organism's brain, skin, and voice. The brain, WordPress 2.6, interprets requests, makes decisions, and coordinates the activities of all parts of the organism. The skin bounds and identifies the organism and non-verbally communicates attitudes and emotions; in Figure 18, two skins, Gear 1.2.4 and Pixeled 1.9, are tried on. The voice verbally communicates facts and ideas; the person who controls the voice is considered the blogger, the author of the blog. Table 3 identifies in detail the three layers of authorship which I consider as relevant to the construction and versioning of the blog illustrated in Figure 18.

Table 3: A blog's brain, skin, and voice are separately authored and versioned.

	title	**version**	**date**	**primary author**	**primary language**
brain	WordPress	2.6	2008-07-15	WordPress.org *team*	PHP (generates HTML)
skin	Gear	1.2.4	2009-02-18	MyMobiles.com *team*	PHP (generates HTML)
	Pixeled	1.9	2009-02-19	“sam07” *individual*	
voice	Hypertextual Ultrastructures	-	2008-12-15	“rose” *individual*	English

The difficulty of correctly and completely describing this blog’s version is caused by its multi-level construction. The blog’s software and themes are versioned in the ways (discussed in detail below) that are usual for software. WordPress 2.7.1 was released February 10, 2009, so at WordPress 2.6 the blog is somewhat back-leveled, exposed to errors that may have already been corrected in the time since 2.6 was released in July 2008; the blog’s brain is not as smart as it could be. Both the “Gear” and “Pixeled” themes, though, use up-to-date levels, last modified in February 2009; each of the blog’s available skins is as good as it can be, barring customizations. Switching between the installed themes changes many aspects of the blog’s appearance and function, but it doesn’t change the extent to which it is current or obsolete. The extended silence of the blog’s voice, however, does mark it as inactive. Blogs were popularized as “the format for rapid-fire diaries of a day’s activities and thoughts” (Bausch et al. 28); a blog that, like this one, has had no change to its content for more than two months is likely to have been abandoned.

The blog in this example is inactive, using software that is slightly but not severely back-leveled and a perfectly-current theme. Upgrading to the newest version of software or a theme, or changing to a different theme, or even replacing the WordPress software with one of its competitors such as MoveableType, does not alter the publication date of the blog's latest posting: the voice is silent, no matter how much the brain and skin may have grown and changed since it last spoke. Because a blog is designed to change rapidly, the blog's readers describe and navigate it like a daily newspaper, by reference to postings' timestamps and headlines but not to any blog-wide version identifier. However, while a human reader of the blog imagines reading the December 15th posting, that posting is managed in the WordPress database not by its date but by a sequence number: www.scribionics.com/blog/?p=36, the 36th page added to the blog, contains the words that were added December 15th.

Texts that human readers perceive as moving slowly can be identified holistically, in ways that are not tied to dates and times but to simple sequencing: a book can have a second edition; a film can continue in Part II; a professional journal can issue a new volume each year and a new issue every quarter; software can be released as a new version, numbered one digit higher than the version it is meant to replace. For programmers, webmasters, and others interested not in the content of the blog but in the condition of its containers, a usable description of the blog necessarily refers to version numbers: the example is a WordPress 2.6 blog using a Gear 1.2.4 (or Pixeled 1.9) theme.

In combination, what the blog *is* (its brain, central and essential) and what it *uses* (its skin, external and ephemeral) describe and explain how it works, how it differs from other blogs, what defects it has, and what repairs it is likely to require. This two-part version identification is not affected by changes in the blog's content; it can, however, be useful in understanding and correcting errors experienced by the author or reader of that content.

Software Creates by Describing

Blogs, like other digital hypertexts, are mediated by software; software and its derivatives are easy to change. To a large degree this is simple physics: because software is built from language and linguistic symbols, not from steel or copper or wood, using different words, or using the same words differently—re-ordering paragraphs, or adding or deleting contextual information—creates a different tool, useful in different ways than the tool it replaces. Just as human-language statements like "I promise" perform the action they name, computer-language statements like "<table width="125">" create the situation they describe. In software, changing the language that describes the tool changes the nature of the tool—how it is shaped and how it can be used—directly and immediately.

Unlike computer languages, human languages consist largely of words which describe but do not create reality. Writing "dining table" or "coffee table" on a furniture inventory has no effect on the presence or absence of actual dining tables or coffee tables in the room that is being described. A textual change to software, however, does alter its own reality: changing "<table width="125">" to "<table width="50">" creates a table with a different width rather than merely describing the width of a table that might exist in an actual or potential external reality.

Understanding this perceived distance between text and reality is essential to understanding the differences between human and computerized textualities. Human textuality ordinarily separates text from the reality it describes; we are aware of a distance created by representation, of ideas and objects and actions existing independently of the symbols that allow us to describe and discuss them. Computers, though, experience reality as created by and inseparable from its description. When humans are able to experience textuality as computers do, not as a representation of reality but as reality itself, the resulting experience of thought as action and change as immediate is what has been called "virtual reality," as discussed in Marie-Laure Ryan's 2001 *Narrative as Virtual Reality: Immersion and Interactivity in Literature and Electronic Media* and elsewhere.

Software Is Language Designed to Change: Versioning, About, README

Within the tool-making culture of software developers, techniques and traditions have developed to support the expectation that software will change easily and often, and that the planning and implementation of changes will be public and detailed, "open to the point of promiscuity," as Eric S. Raymond describes it in "The Cathedral and the Bazaar" (21). Because software versions are conventionally identified with multi-level sequential numbers, the expectation that software's versioning describes not only the fact but the degree of change between one version and another is a source of ongoing debate among developers about whether a particular software change requires increasing the top level of the version (Version 1 becomes Version 2) or some lower level (Version 1 becomes 1.1 or 1.0.1). For example, the open-source Kompozer authoring system has a well-documented multi-year revision history but identifies its current level as Version 0.7.10, defended on Kompozer's "About" page at kompozer.net/about as follows:

> Because for 99% of the developers in the world, including myself, a «1.0» version means it's ready for professional use. [...] there are some bugfixes and features that are necessary before I can decently call it a 1.0 version.

Software versioning makes a specific claim; in the example above, claiming to be Version 1 ("ready") is so different from claiming to be Version 0 (developmental) that the change cannot be "decently" made until specific criteria are met.

Because fully identifying a software tool, including its version, is a prerequisite to obtaining help with that software, any example of modern software will almost certainly identify its own version with a "Help—About" process yielding something resembling the examples shown in Figure 20. Figure 20 shows the "Help—About" responses for two competing word processing tools: Microsoft's proprietary Office Word, on the left, and Sun's open-source OpenOffice.org, on the right. While these "Help—About" windows are formatted very differently from each other, reflecting the strong differences between their sponsoring organizations, they both perform the key functions of a "Help—About" response. Both provide the essential facts of the software's complete identity: the version of Word installed on this computer is 11.8169.8172, with

Service Pack 3 added; the version of OpenOffice.org, a much younger tool than Word, is 2.2.1.

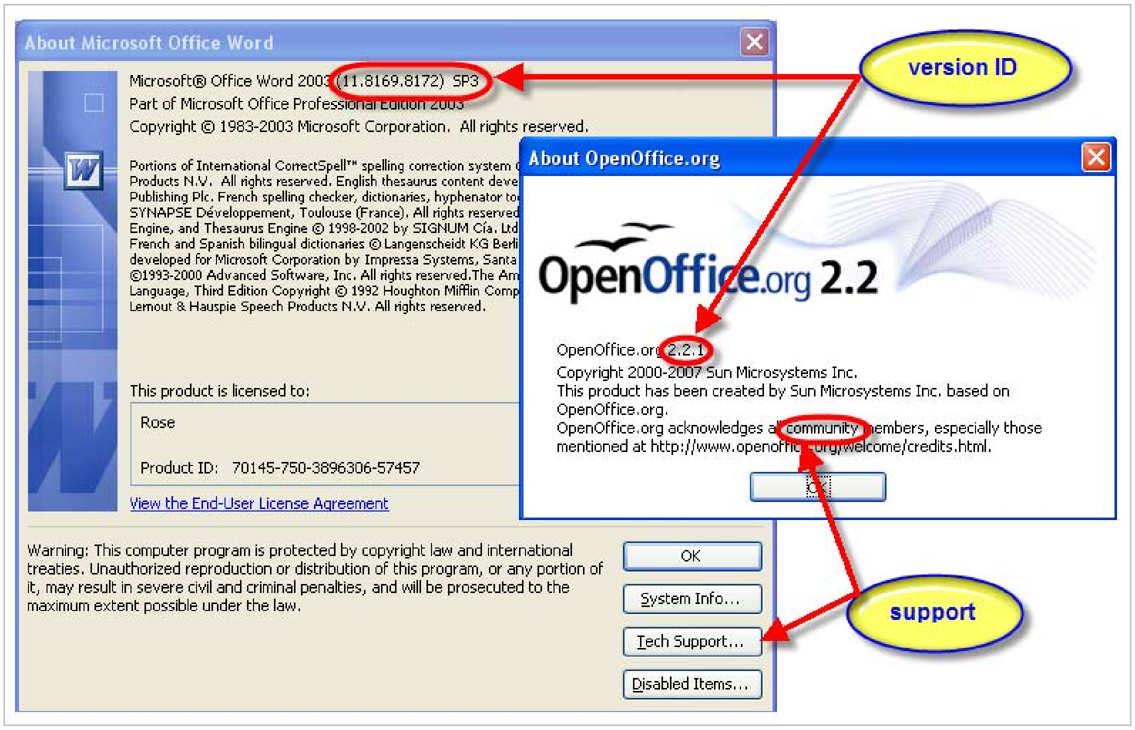


Figure 20: Conventional Help—About responses identify the software version and a source of support.

For both Word and OpenOffice.org, "Help—About" also suggests sources of support, although those sources also differ strongly. For Word, clicking the "Tech Support" button presents a summary of resources, "subject to then-current prices, terms, and conditions, which are subject to change without notice." For OpenOffice.org, software named identically with the Web site at which it is published and supported, mention of the "community" is a reminder that this is open-source software, created and supported by volunteers who can be reached at www.openoffice.org. With these two sets of facts in hand—the complete identity of the software, and contact information for people who are knowledgeable about it—even an untrained, seemingly-solitary user can expect to benefit from the collective knowledge of the software's authors.

Another element of the software versioning tradition, the "Readme" file, addresses the person who installs the software rather than the person who uses the software. Although on a home-based personal computer the installer and the user of software are likely to be the same person, the "Readme" tradition originated in the days before personal computing, when specialized technical support personnel were expected to handle software installation and configuration; in large organizations, such as businesses and universities, the distinction persists and the assumption of an expert readership remains reasonable. Even in the absence of any specific instructions to do so, an experienced software installer seeks any document named "Readme" (or some near variant such as README, ReadMe or Read-Me-First) among the possibly-thousands of files associated with a software installation; such a document, as shown in Figure 21, provides detailed information about how this version of the software differs from its predecessors and how, therefore, its installation and operation also differ.

```
-------------------------------------------------------
Contents:
-------------------------------------------------------

- What's New in Norton Internet Security 2007

- System Requirements

- Unblocking applications that have been blocked by the Norton Internet Security Firewall

- Installation on removable drives, FireWire drives, etc.

- Installation on multi-boot systems

- Installation on computers running server services

- Order of installation for Symantec products

- Uninstalling Norton Internet Security

- Upgrading operating systems after installing Norton Internet Security

- Browser Compatibility

- ICU License

- Accessibility - High Contrast

- Documentation Addendum: Adding computers to Trust Control

- Documentation Addendum: Spanish Language Support

- Other Third Party Acknowledgements
```

Figure 21: Portion of a README file.

In this readme.txt file provided with Norton Internet Security, changes in the software itself are identified in the “What’s New” section; other portions of the document address changes related to external factors such as new types of hardware (including FireWire drives), expanded choices of Web browsers (including Opera and Safari), and broader inclusion of human users (high contrast for users with visual impairments; Spanish language support). Pre-installation examination of the Readme provides early warning as to what changes will require responses (such as hardware changes or additional user training), but such documents are also intended to operate as reference material throughout the useful life of the software.

Software Is Language Surrounded by Language

Software operates within a supporting environment of language, including installer-oriented configuration information such as Readme.txt and user-oriented operational information such as Help—About. Before it is made available to installers and users, software is constructed within an even more extensive collection of texts, including user requirements, design specifications, error reports, validation plans, and testing results. The existence, and to some extent the format and content, of the texts that surround and support software development is due in part to practical reasons related to coordination of large teams working on long-term projects, and in part to the government-enforced requirements of industries such as banking and medicine which require extensive evidence that the software tools they use are reliable.

The texts that surround software are also created and maintained for traditional reasons grounded in the mores of the tool-building culture of engineers and programmers, who see their work as multi-generational and value accurate record-keeping as a source of aid to their intellectual descendents; writing "for the record" is directed toward a cadre of imaginary future replacements, keeping software in use by supporting rather than resisting change. In these circumstances, creation of texts that not only *are* the software but explain how to use it and demonstrate that, in the face of change, it remains useful and true to its design, are not items of curiosity but are required for basic survival. Even a single software module exists at the center of a cloud of other texts; most of the texts surrounding software are not intended to be accessible to its ultimate users. In Figure 22, I have identified some of the most common forms of texts created in association with software, grouping them into spheres identifying their most likely readers.

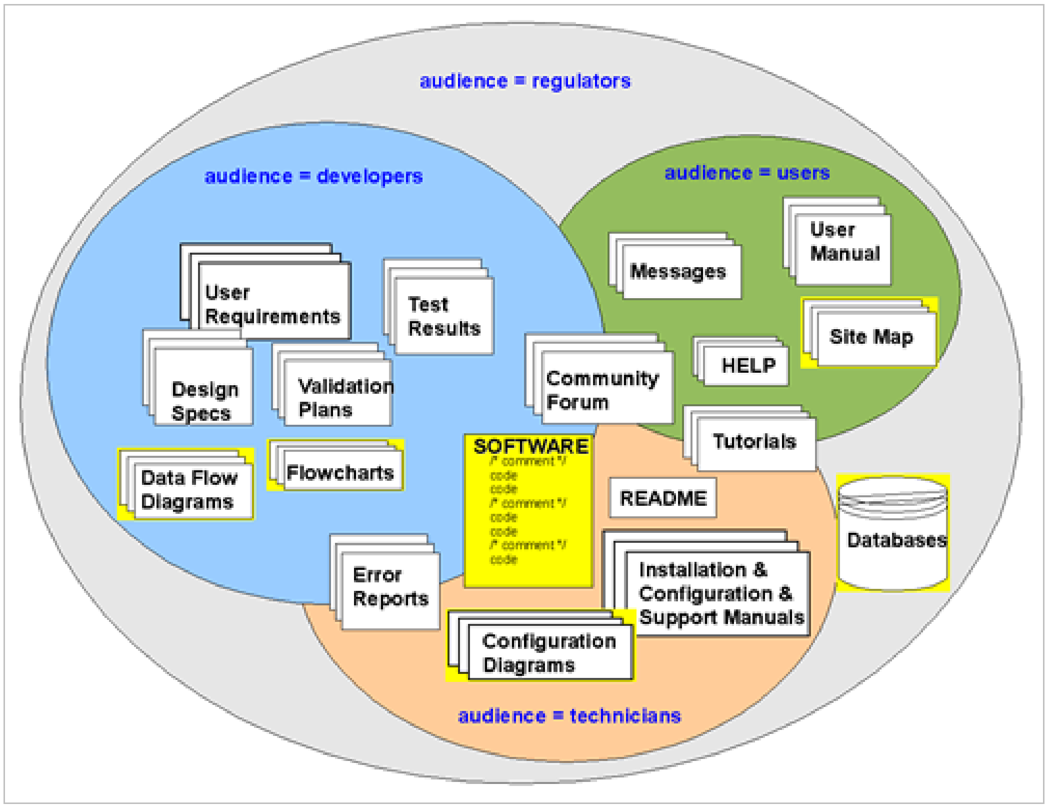


Figure 22: Software within its context of supporting documentation.

Here, items framed in yellow are graphical (data flow diagrams, flowcharts, configuration diagrams, site maps) or are created in artificial languages (databases, the software itself); the other items, including comments imbedded within the software, are created in human languages. The relative sizes of the spheres in Figure 22 suggest the volume of software-supporting texts available to each audience; members of each audience may also have opportunities to create new texts about the software.

- Users, interacting with software at its surface level, have access to related *instructional* texts such as help files and error messages; if the purpose of the software is to create new texts such as blog postings, they may use it to do so; whatever the purpose of the software, its users may be able to create texts discussing it and asking for help by means such as community forums.
- Technicians, caretakers responsible for installing and maintaining software as well as for direct support of its users, rely on *descriptive* information about the software's construction and the ways in which it can be customized for its intended uses; they are likely to create internal-use texts documenting their configuration and installation histories, as well as public texts answering users' questions.
- Developers, the software's authors, create not only the software itself but their internal-use *historical* texts recording how the software was designed and tested; perhaps with the support of specialists in technical communication, they are likely to have also created the descriptive and instructional texts provided to technicians and users.
- For software used in regulated industries such as banking and pharmaceuticals, regulators must have access to all the texts created for and by the other audiences, using them as *evidence* that the software is or is not created, supported, and used according to applicable regulations; the regulations themselves, of course, also exist in texts external to the software, as do regulators' audit findings and other reports.

Wrapped in their thick blankets of self-description, the intentional differences between one version and another of a software tool are likely to be completely obvious and thoroughly detailed; other texts, including software-mediated hypertexts, are not customarily accompanied by layers of self-description, leaving the human reader to investigate the differences among multiple versions. Direct comparison, searching for textual variants, can be an effective way to do this for printed texts; however, the results of a comparison become less usable when the differences relate to structure, function, or appearance rather than to the text's linguistic content. In what follows, I will demonstrate some methods by which differences can be identified and visualized on the basis of structural and ultrastructural changes.

Software Is Made of Words but Explained by Pictures

Most software languages are verbal; programming languages that express ideas with images rather than words do exist, but are primarily tools for teaching and experimentation, and even "visual" languages—Visual C++, Visual Basic—are so named because they are designed to simplify the presentation of visual information and not because their syntax is less verbal than that of other languages (Kahn 50). As Thomas G. West argues in "Visual Thinkers and Nobel Prizes," words are not the only form in which humans can receive information, and are indeed not the preferred form of communication for most programmers and engineers; a variety of non-verbal methods have been developed as methods of designing, debugging, documenting, and evaluating software.

Because I think many texts, especially hypertexts, can be usefully interrogated in many of the same ways that are applicable to software, I also think development of visualization tools suited to communicating the structure, behavior, and evolution of non-software texts is essential. The stemma, graphically presenting a text's place in a "continuous chain of descent from the original source" (Tarrant 104), has long been the primary aid to visualization within textual studies. Like any other family tree, a stemma can present two facts about the relationships within a set of texts: chronological order, visually separating older and newer texts; inheritance, visually connecting parent and child texts. In understanding the evolution of a hypertext, the facts of order and inheritance among multiple versions are no less worth knowing. However, a more useful visualization of such multi-layered texts would itself be multi-layered, showing how versions can differ from each other in matters of their internal structures even when their linguistic content appears to be unchanged.

For two versions of a hypothetical hypertext, identical in surface-level content but not sub-surface structure, Table 4 suggests eight criteria on which both versions' ultrastructures might be compared; these are eight possibilities selected from a much larger universe. For each criterion, the table includes a short explanation, along with numerical weights comparing the pervasiveness of each criterion in each version.

Table 4: Numerical comparison of two versions of a hypothetical hypertext.

criterion	Version A	Version B
Comments: internal explanations, addressed to hypothetical future humans who wish to understand the reasoning behind the construction of the page	70	10
Metacode: internal descriptions, addressed to search engine robots to improve the page's likelihood of being identified appropriately as a search result	70	40
Stylesheets: formatting is defined once	70	20
Local Images: images are stored at the same site as the page	98	62
Alternate Text: support for non-visual browsers	90	80
Dynamic: HTML is generated on demand	0	100
Scripted: logic is defined once	80	90
Scalable: sizes are expressed as percentages of the screen size	50	50

While comparison of some of these criteria can be the basis of assumptions about the relative ages of versions, such assumptions can be only tentative without additional supporting evidence. For example, the technique of coding dynamic pages, using a programming language such as PHP to assemble HTML in response to user input, is newer than the technique of coding static pages, with multiple similar pages pre-defined to meet anticipated possibilities; if all else is equal, it can reasonably be argued that the version weighted "100" for the "Dynamic" criterion is younger than the version weighted "0". The prevalence of other coding practices has also changed with time: increased expectations of attractive formatting have created increased reliance on stylesheets; growing commercialization of the Web has increased the need for search engine optimization through use of metacode. Awareness of such changes in hypertexts' enabling technologies is as useful in relating hypertextual versions to each other as awareness of changes in printing technologies—availability of a new kind of paper, for example—can be in placing printed texts in an appropriate chronological relationship to each other.

Comparing two versions in terms of eight ultrastructural criteria, a complex task, is made more complex not only because hypertextual versions can evolve rapidly, but because what can appear to be single criterion for evaluating the sub-surface nature of a hypertext can itself be examined on multiple levels. For example, as will be discussed in the "Various Kinds of Variability" section of Chapter V, identifying a Web page as dynamic is in itself only a high-level label: there are many ways in which a dynamic Web page can be dynamic. Given that a full description of the commonalities and differences between two versions can require multiple simultaneous comparisons, summarizing the comparison with a drawing such as Figure 23 can be a more useful expression of the relationship between two hypertexts than a table or list of the (potentially very many) ultrastructural details in which they differ. In Figure 23, the shape representing Version A is shaded light blue, with a horizontal hatching pattern; the shape representing Version B is shaded red, with a vertical hatching pattern; by combining their colors and hatching patterns, the central shape represents the extent to which the versions are similar.

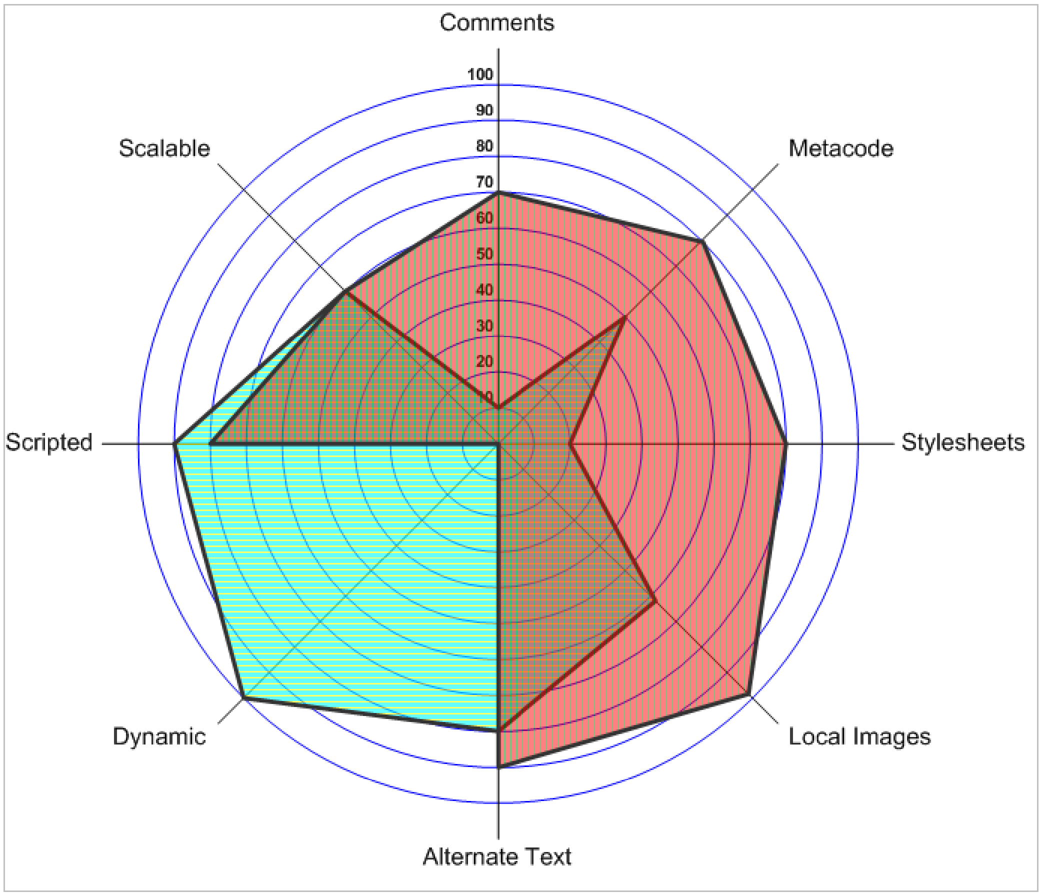


Figure 23: Spider diagram comparing two versions of a hypothetical hypertext.

In *Software Visualization*, Diehl recommends the spider (also known as radar or Kiviat) diagram as an efficient method of visually summarizing multiple differences between versions of a software module. Spider diagrams are easily drawn, by hand or with the aid of readily-available tools; the example above was produced with a Microsoft Visio template. Spider diagrams are also widely used to support planning and evaluation of IT projects: creating spider diagrams of several alternative configurations and comparing them side-by-side provides an easy method of communicating their similarities and highlighting possible problematic areas of difference. Figure 23 visualizes the same data as Table 4: eight ultrastructural criteria by which to compare two versions of a hypertext. In this spider diagram, areas of analysis relevant to comparing ultrastructural differences are assigned to the radial spokes on the spiderweb, with ascending values indicated by distance from the center of the web; for each version of the hypertext, placing a mark on each spoke to indicate its level within that version, then connecting all the marks into a closed polygon, creates a shape that summarizes the hypertext's condition in that version; repeating the process for multiple versions and comparing the shapes produced visually highlights the areas in which they overlap and differ.

In one image, the spider diagram communicates that the ultrastructures of these two versions of the "same" Web page are vastly different in one way (A is a dynamic page; B is static), minimally different in another way (B provides alternate text for a few more of its images, making it slightly more usable by visually-impaired readers than is A), and identical in another way (there has been no change in the scalability of tables, making A and B equally capable of adapting to varying screen sizes). When, as in Figure 23, both versions have increased their use of some but not all new practices, a nuanced explanation is called for. Perhaps both versions were independently developed from an unknown original, with each group of developers prioritizing different ultrastructural changes; perhaps the version that seems newest on some spokes but not on the "dynamic" spoke was once also dynamic, but the developers encountered some difficulty in that aspect and abandoned it, returning to static coding techniques. A spider diagram can only present the facts of differences between versions; interpreting those facts remains a job for curious humans.

Describing Intentional Change in Informational Works

Where a stemma places a text in relation to other texts of the same work, a textual description describes a single text in sufficient physical detail—the signs that guided assembly of physical sheets of paper into the book's pages; the locations of illustrations; the printer's imprint—that this text can be identified as an individual among other texts of the same work. At a deeper level of detail than this book-wide structural description, textual notes point out ways in which texts vary internally. These variations can be caused by each printer or publisher's consistency with their own house style in matters of spelling and punctuation: "dining parlor" and "dining parlour" and "dining-parlour" and "dining-parlor" are functionally identical (Opperman 100-101), differing in hyphenation and in use of British or American spellings, but not in the ideas they are meant to convey. Tracking variations of this kind are useful in tying a printed text to one printer or another and in establishing some understanding of the kinship relations within a group of texts; for imaginative works, such minor intentional variation may spark investigation by scholarly editors who may eventually determine what the author, irrespective of any printer's house style, preferred.

For multiple editions of informational works, though, sorting through spelling variants is not likely to be useful as a step toward understanding authorial intent: the author intends the text to present current information, and as such information is subject to constant change then so is the text, in ways much larger than minor adjustments to spelling and punctuation. Other tools are needed to describe and explain the ways in which texts incorporate major changes, including reordering of sections, addition of new material, and deletion of obsolete material. The handling of obsolete material in informational texts is particularly problematic, since it is often necessary to preserve the obsolete information, using it as the basis of explaining what has changed, rather than simply deleting it.

Visualizing Revisions of Procedural Information: All Versions Influence All Versions

For technical documentation, any visualization of the relationships among revisions must account for the reality that earlier versions of the document are presumed to have a permanently valid existence (archived for legal purposes, recording the instructions under which people were operating at a previous time, if nothing else) and to influence all later versions. Figure 24, for example, depicts the paths by which five versions of the same document influence each other.

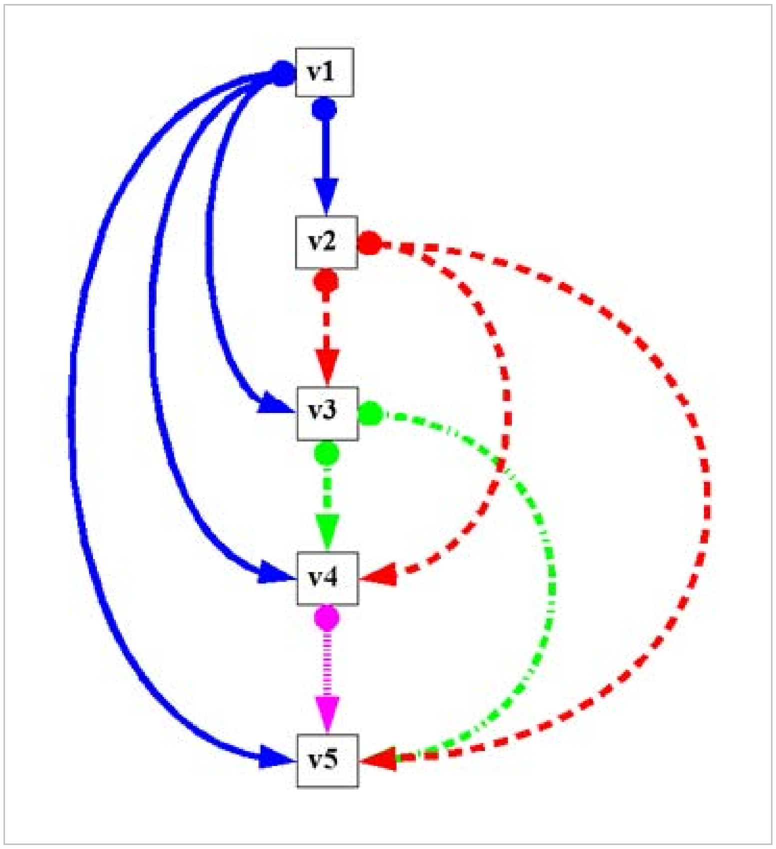


Figure 24: For technical documentation, a whirlpool diagram shows multiple possible lines of descent.

In what I call a “whirlpool diagram” such as Figure 24, the influences of all earlier versions on all later versions are acknowledged. There is a simple, direct line of inheritance among versions: as represented in the central, downward-pointing line, Version 1 is the primary source of Version 2, which is the primary source of Version 3. However, the direct chronologically-ordered line is not the only line available; every obsolete version may provide source material to every new version. This is especially true because technical documents must remain consistent with a changing reality, and reality can change in multiple directions. A feature that was included in Version 1 of a software tool must be explained in Version 1 of the software's documentation; if that feature is eliminated in Version 2 of the software and then, in response to user complaints, restored in Version 3, the matching documentation must change in the same ways. Wikis implement this digitally, with the "revert" function making it possible that the most-recently-written text is not the currently-used text; a whirlpool diagram showing lines of inheritance among versions of a wiki page might, after multiple revisions and reversions, look more like a series of cascading spirals than like a single whirlpool.

In addition to returning to an older explanation because an external reality has returned to an older practice, a new version is also likely to incorporate portions of an older version for the purpose of elaborating on the implications of the change, especially in terms of how users must change their long-standing procedures: because the software has changed, the documentation and the users' activities must also change. This fundamental assumption that readers of the current text are likely to have expectations created by an obsolete previous text, and that a major purpose of the text is to help those readers adapt their pre-existing behavior to the changed reality, is an important difference between works of information and works of imagination. Detailed comparisons of versions, under headings such as "What's New in This Release?" or "Summary of Changes," are typically highlighted in the introductory sections of software user's guides, instructions associated with government forms, and other documents that are closely tied to a changing external reality; in such documents, textual differences are meant to be highly obvious, creating little occasion to exercise the skills of a textual scholar in closely comparing multiple versions and reporting on how they differ from each other. For such documents, failure to explicitly announce how one version differs from another can lead the document itself to be considered a failure, unusable because it is not clear what portions of the text must be re-read by someone who previously *used* (the purpose of such documents being to *use* their information) another version.

Another essential feature of informative procedural texts is that they must support being read as fragments, as the information in each fragment is needed, and their structure—highly granular, and supported within a framework of finding aids such as indexes, section headings, and tables of contents—is created to shorten, rather than to prolong, reading time. In a good informational text, a reader can easily determine whether a required piece of information is present, and then quickly locate that information and, applying the explanations and illustrations provided in the text, put the information to use; there is no expectation of extended contact between reader and text. Such a text would be considered to have failed if extensive sequential examination (reading) rather than direct access (searching) were required before information could be extracted.

Visualizing Versioning of Paper-Like Texts

Non-procedural informational texts describe and discuss reality without instructing readers as to how to use or alter reality. Rather than explaining how to *do* something—install a printer or complete a tax form—such texts focus on explaining how to *understand* something. Reference material such as dictionaries and encyclopedias are in this category, operating somewhat similarly to procedural works in that their structures are designed to support rapid, fragmentary reading and minimal contact-time with readers. Other non-procedural works of information are designed with some expectation of sequential reading but strong support for direct-access reference; many scholarly works such as monographs and textbooks are in this category. Informational works of this kind are also occasionally re-published in new versions as new information is developed.

When they are found on the Web, works in this category often retain characteristics of printed publications: a scholarly journal's Web site may offer access to journal articles as electronic paper (.PDF); a library's online catalog may include an e-book rather than a paper book. Methods of identifying and visualizing intentional changes for these paper-like works are needed; however, there being no equivalent of View—Page Source in the tools such as Adobe Reader that enable viewing of electronic paper, the internal structures of electronic paper are not accessible in the ways they are for other hypertexts. For digital texts constructed like software, internal structures can be examined and visualization methods such as spider diagrams, appropriate for comparing software versions, may be applicable; for digital texts constructed like paper, only surface layers can be examined and visualization methods appropriate for paper are a better fit. For texts published on paper or electronic paper, the surface-level content of multiple paper-like versions can be used as input to automated visualization tools; the resulting images can clarify understanding of differences among the versions. However, because the differences between versions can be differences in organization as well as in content, it is important to compare the order in which ideas are presented, not simply the easily-tallied facts of the inclusion and repetition of words. Changes in structure can be usefully visualized and compared for multiple versions of any text which describes itself with a Table of Contents or some analogous navigational aid. In what follows, I demonstrate the usefulness of two visualization methods: tag clouds, for which a variety of image-generation tools are available; bead diagrams, which are my own design and must currently be modeled by hand or with the aid of general-purpose drawing software.

Case Study: George P. Landow's Hypertext, Hypertext 2.0, and Hypertext 3.0

Using a non-procedural work of information that exists in multiple versions, George P. Landow's *Hypertext* (1992), *Hypertext 2.0* (1997), and *Hypertext 3.0* (2006) books, I demonstrate two methods of visualizing the differences among them. My interest is not in addressing the content of these books, but in using accessible elements of their structures to explore versioning in these paper-like, non-procedural informational texts. Because sub-surface layers are not available in paper-like texts, analysis of their evolutionary changes must begin with any self-description provided with each text. In what follows I develop multiple visualizations based on each version's self-descriptive Table of Contents. Tag clouds illustrate the evolution of the work's subject matter; bead diagrams illustrate the evolution of the work's structure. More efficiently than a lengthy verbal description of changes, these visualizations identify the ways in which the versions differ from each other.

The books themselves provide little detail about the changes that prompted new versions. *Hypertext 2.0* includes no comment as to how it differs from *Hypertext*. In

Hypertext 3.0, the first paragraph of the Preface summarizes the changes between *Hypertext* and *Hypertext 2.0* as replacements (drawing examples from the Web rather than Brown University's internally-developed Intermedia) and insertions (adding material on "writing for e-space," new hypertext fiction, "and so on"); in the same paragraph, the differences between *Hypertext 2.0* and *Hypertext 3.0* are identified as five-fold, adding discussion of the Web's growth, blogs, animation, globalization, and digital cinema (xi). In combination, these brief statements set only general expectations about the ways in which these versions were intentionally changed. The books are not software, so it is not reasonable to expect the level of detailed self-description that is usual in software versioning; however, the method of identifying their versions, labeling them as "2.0" and "3.0" rather than "2nd edition" and "3rd edition," suggests that these texts are in some way software-like or that they could be read as in some way similar to software. Hypertext, their primary subject matter, is indeed intimately bound up with software; developing some details about how these versions differ from each other may clarify what is meant by the software-like titles of these paper-like books.

Dino Buzetti and Jerome McGann, in "Critical Editing in a Digital Horizon," identify comparative analysis as "the basis of all scholarship" (58). I think there are non-comparative activities that are also fundamental and valid, describing what a text *is* rather than how it *differs* from another text, but I agree that comparison is a necessary and practical task with clear, usable results. Because comparison yields concrete, measurable data, it lends itself ideally to visualization as a means of efficiently communicating the differences among texts. However, complete textual comparison is more practical for small texts such as sonnets than for long texts such as scholarly monographs. Comparing these three book versions character-by-character is not useful: not only has new material been added, but existing material has been removed or re-ordered within this large (*Hypertext 3.0* has 436 pages) work; an accurate list of variants would itself be book-length, too detailed to directly answer questions about how the versions differ from each other.

Since comparing the full texts provides too much information and the author's description of their differences, including their confusing titles, provides little guidance, the ideal level of focus is on the structural level between them: their Tables of Contents. I use the Tables of Contents in two ways. First, by generating one tag cloud per version, I create a single-image summary of each version; comparing the tag clouds creates a holistic understanding of the ways in which, by identifying their subject matter in their Tables of Contents, the versions claim to differ from each other. Second, using the same Table of Contents data, I create bead diagrams to track the movement, reordering, and deletion of structural elements as the text evolves through its three versions; the results highlight areas of commonality and difference in the organization of the versions.

A *tag cloud* (also known as *word cloud* or *weighted list*) generates one image representing the frequency of word use within a text; on the Web, it is widely used to visually summarize the current state of a single, multi-topic digital text such as a blog or forum. Comparing tag clouds for each of several revisions of a text makes it possible to observe, from among the many possibly-trivial ways in which the texts differ, where the substantial changes are. This is what I have done for the three versions of *Hypertext*.

From the books' three Tables of Contents, including the titles of chapters and subsections, I created plain text (.txt) files; using those files as input to the tag cloud generation software at www.tag-crowd.com (Steinbock), I generated one image to represent each version of *Hypertext*. The three figures below present the resulting images in chronological order: Figure 25 is the tag cloud for *Hypertext [1.0]*, followed by Figure 26 for *Hypertext 2.0* and Figure 27 for *Hypertext 3.0*.

Figure 25: *Hypertext [1.0]* weights "hypertext" and "text" similarly.

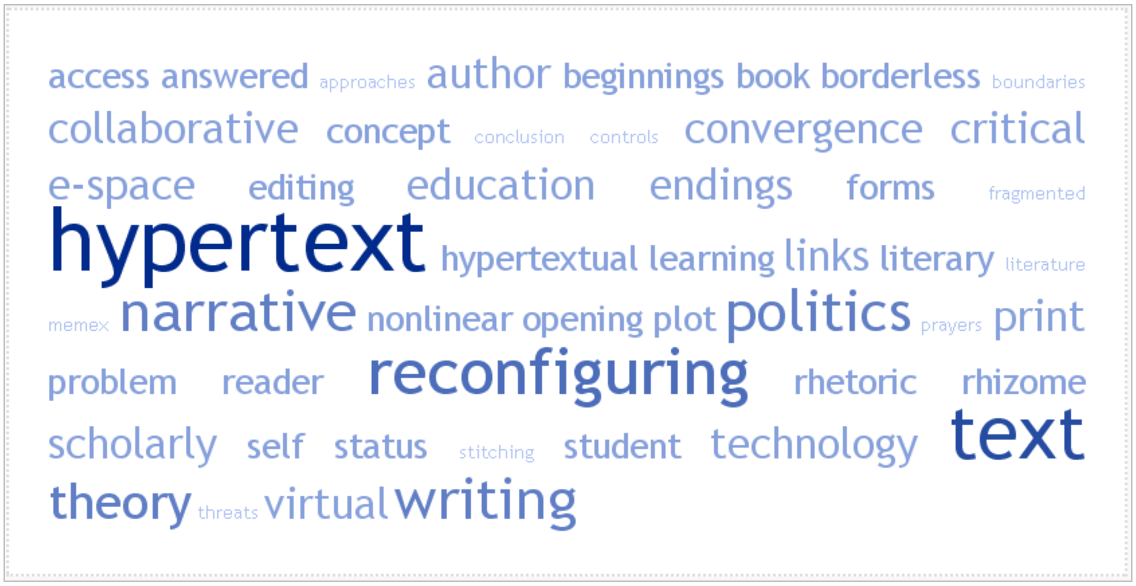


Figure 26: *Hypertext 2.0* weights "hypertext" most heavily.

Figure 27: *Hypertext 3.0* widens the gap between "hypertext" and all other ideas.

For each book's Table of Contents, the tag cloud lists alphabetically the fifty most frequently used words, with similar words aggregated and common non-meaningful words ignored. In all three tag clouds, "hypertext" is the most heavily weighted, as it should be. The weights of other less-central words change noticeably:

“collaborative,” for example, is an important matter in the first version, somewhat less important in the second, and much less so in the third; “critical” and “reconfiguring” follow the same pattern. “Boundaries” is visible as a minor matter in the first two versions; the word is not in the top fifty for the third version. The first two tag clouds share a pattern of heavily emphasizing many of their top-fifty words. In the third tag cloud, fewer large, bold words indicate that more repetition has been focused on fewer words, an obvious break from the earlier pattern: the word frequencies visualized by the tag clouds identify *Hypertext 3.0* as measurably different from the earlier two versions, which are not identical but strongly resemble each other. Based on the tag clouds, these versions might reasonably be described as *1.0* (the first published version), *1.1* (a minor update), and *2.0* (a redesign).

Tag clouds are based solely on the presence of words, not on their sizes, colors, fonts, or positions. This is the purely-formal way a computer reads, weighting the words as words without concern for cultural matters that might alter a human reader’s perceptions: a human might be expected to attach greater weight to a word that appears once in a 24-point bold red heading than to a word that appears five times in a 10-point normal black font in the body of a text; for a computer, such an interpretation would be most unnatural.

The tag clouds compare the ways in which, through their Tables of Contents, the three books identify the ideas they discuss. While comparing the ideas they contain is important in understanding the development of the three versions, comparing the order in which those ideas are presented is also useful. To do this, I have created two versions of what I call a "bead diagram," visualizing structural changes as described in the evolving Tables of Contents. In a bead diagram, structural elements (such as front matter, book chapters, and back matter) in the original version are visibly tied to their re-instantiations in later versions, showing cross-version structural modifications such as insertion, repositioning, or loss. Rather than feeding electronic versions of the Tables of Contents into image-generating software as I did to produce the tag clouds, I drew the bead diagrams myself; I used OpenOffice.org's Draw tool to create the first diagram and Microsoft's Office Visio to create the second. Figure 28 visualizes the structure of the top-level Table of Contents entries.

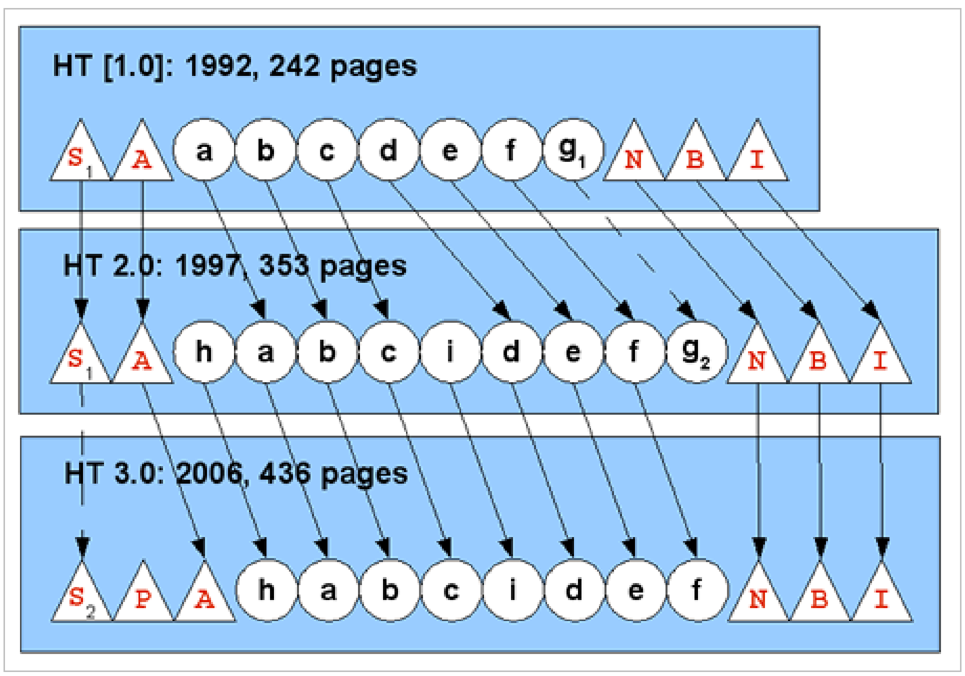


Figure 28: Bead diagram tracks changes in structure among three versions.

The “beads” here consist of shapes indicating the type of structural object represented: triangles represent paratextual apparatus; circles represent book chapters. Each bead is marked to label its content and positioned horizontally in the order it occurs within each version; vertically, each bead is connected with an arrow to its re-instantiation (if any) in the next version. Even without naming the Table of Contents entry represented by each bead, areas of structural change can be identified: for example, the round bead labeled “a” appears in all three versions, but it is pushed away from the beginning of the string as introductory material is added in each version. Table 5 matches each bead to its identity in the Tables of Contents.

Table 5: Bead labels match section names in Tables of Contents.

bead	TOC section name
S_1	book subtitle, “The Convergence of Contemporary Critical Theory and Technology”
S_2	book subtitle, “Critical Theory and New Media in an Era of Globalization”
A	front matter, “Acknowledgments”
P	front matter, “Preface”
N	back matter, “Notes”
B	back matter, “Bibliography”
I	back matter, “Index”
a	numbered chapter, “Hypertext and Critical Theory”
b	numbered chapter, “Reconfiguring the Text”
c	numbered chapter, “Reconfiguring the Author”
d	numbered chapter, “Reconfiguring Narrative”
e	numbered chapter, “Reconfiguring Literary Education”
f	numbered chapter, “The Politics of Hypertext: Who Controls the Text?”
g_1	numbered chapter, “An Open-Ended Conclusion; or, The Dispatch Comes to an End”
g_2	unnumbered section after final chapter, content similar to g_1
h	numbered chapter, “Hypertext: An Introduction”
i	numbered chapter, “Reconfiguring Writing”

Based on the high-level structural changes shown in Figure 28, both later versions are minor updates, adding some material but not altering the original's fundamental approach. However, this bead diagram compares only the top levels of the Tables of Contents; because the book chapters are divided into named subsections, performing a similar study of the evolution of that deeper level of structure may clarify the ways in which the versions differ.

A bead diagram visualizing the inheritance relationships among the full structures of three large books can produce more detail than can be visualized clearly on standard-size paper, though it may work well on large map-plotter paper or in a digital image which can be scrolled or magnified as appropriate. Within the physical constraints of this dissertation, a bead diagram of a second level of structure for one book chapter can be feasibly presented, as shown in Figure 29.

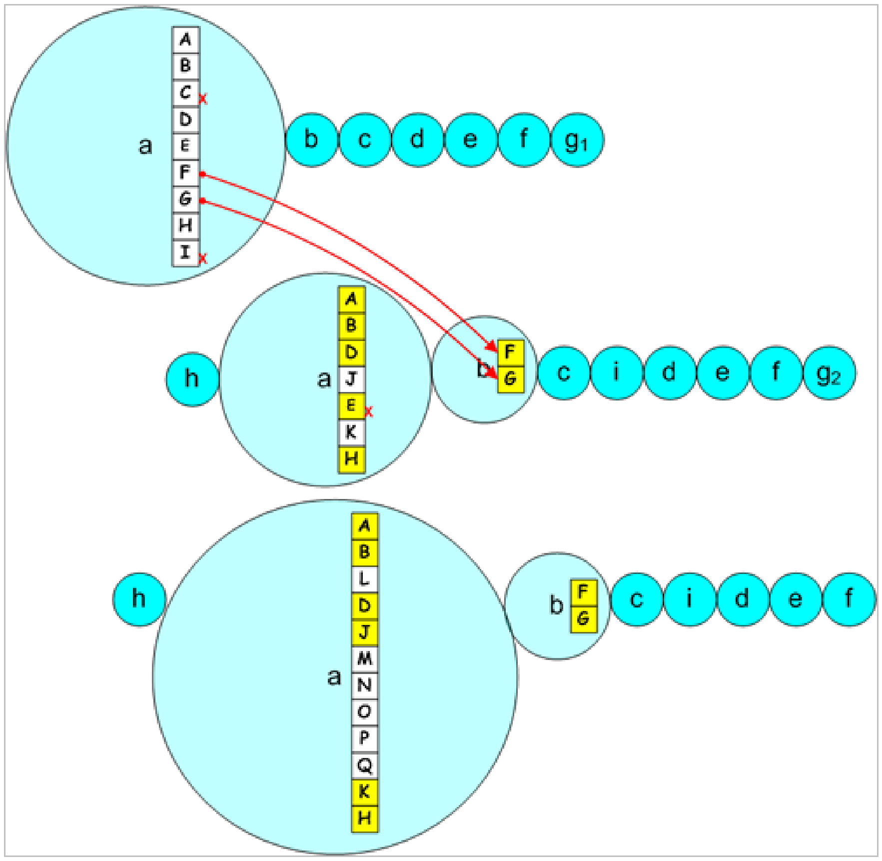


Figure 29: Bead diagram shows re-structuring of contents of original Chapter 1 in three versions.

In this bead diagram, the round beads represent the same book chapters as in Figure 28; the focus here is on the contents of the round bead representing what was originally the first chapter, titled "Hypertext and Critical Theory" and labeled "a." The added level of detail, tracking the movement of named subsections in multiple versions of that chapter, is represented by square beads within the round bead; in a three-dimensional model, those subsection beads should be hanging below each chapter bead. For each version of *Hypertext*, beads representing subsections introduced in that version are white, while subsections inherited from a previous version are colored yellow. For visual clarity, only non-standard inheritance is marked with an arrow; this is the case for the beads labeled “F” and “G,” which begin in “a” but are relocated into a later chapter in *2.0*. Evolutionary dead ends are also visually emphasized with a red “X” beside the bead: two subsections of the original *Hypertext* do not appear in later versions; one subsection inherited into *2.0* from the original is not preserved in *3.0*. Table 6 matches each square bead to its identity in the Tables of Contents.

Table 6: Labels of square beads match subsection names in Tables of Contents.

bead	TOC section name
A	“Hypertextual Derrida, Poststructuralist Nelson?”
B	“The Definition of Hypertext and its History as a Concept”
C	“Other Convergences: Intertextuality, Multivocality, and Decenteredness”
D	“Vannevar Bush and the Memex”
E	“Virtual Text, Virtual Authors, and Literary Computing”
F	“The Nonlinear Model of the Network in Current Critical Theory”
G	“Cause or Convergence, Influence or Confluence”
H	“Analogues to the Gutenberg Revolution”
I	“Predictions”
J	“Forms of Linking, Their Uses and Limitations”
K	“Books are Technology, Too”
L	“Very Active Readers”
M	“Linking in Open Hypermedia Systems: Vannevar Bush Walks the Web”
N	“Hypertext Without Links?”
O	“The Place of Hypertext in the History of Information Technology”
P	“Interactive or Ergodic?”
Q	“Baudrillard, Binarity, and the Digital”

The changing configurations of these beads show that the versions have developed not only by adding material but by deleting and relocating it. It also shows that the steps between versions are not of equal size: of the twelve subsections in the newest version of the “a” chapter, six originated there, two originated in *2.0*, and four were inherited from the oldest version; assuming all the subsections are of equal size, this chapter of *3.0* is 50% (6/12) new material while the same chapter in *2.0* is 28.5% (2/7) new material.

Pursuing bead diagram visualization into the remaining chapters of the books, and possibly into deeper levels of structure within each chapter, would produce a model of textual change as complex and beautiful as models of growing crystalline structures. The bead diagrams included here were produced digitally (by drawing with software) but not automatically (by feeding raw data to software); to automate this process for larger projects, many existing tools for tracking development of databases or software systems may be adaptable. A survey of all potentially-useful visualization software is beyond the scope of this dissertation, and would in fact be useless unless performed immediately prior to undertaking a specific visualization project; this is a highly active area of software development, so the best tool available today may be only a primitive ancestor of the best tool available next year. For further development of multi-dimensional models of textual structure and evolution, it may be that the best available tools come not from software development but from chemistry, where plastic ball-and-stick models (and their software simulations) representing atoms and bonds support even the most advanced investigations into molecular structure.

Multiple visual models add to the sparse description in *3.0*’s Preface of how these versions differ from each other, but they don’t argue for the two later versions as equally-major re-thinkings, despite the “2.0” and “3.0” in their titles. In the same

Preface, Landow credits a family member for the idea of numbering versions in this way, but I think this is worth further interrogation and a suggested interpretation: Landow is from a professional culture (textual scholarship, specializing in Victorian literature) that does not use digital names to formally describe textual versions; he writes about a subject (hypertext) which is largely the product of a professional culture with a strong tradition of versioning in this digital way; to merge both worlds, his titles use names that gesture toward digital practice without indulging in it. This is a move toward surface-level affiliation which may actually do more to identify distance, just as when U.S.-made films about Russia create titles using Cyrillic "Я" in place of Latin "R" (e.g. "DiЯectoЯ of PhotogЯaphy"), even though "Я" makes a sound like the Latin "Y." "Faux Cyrillic" is a common-enough stylistic choice that software such as the "FДКЗ СУЯILLIC GЗИЗЯДТФЯ" is available to automatically produce "linguistic nonsense" with a "Russian or Soviet feel" (Kalilich). I think "faux computing" is also becoming a recognizable style, evocative but not precisely meaningful. Just as "Я" in an English title hints at Russian-ness but does not actually engage in it, the "2.0" in the title of a minor revision hints at software versioning practices without engaging in them. Other computer-ish works similarly gesture incompletely toward digital technology: for example, the cover art of *Race in Cyberspace* (edited by Kolko, Nakamura, and Rodman) uses a pattern of alternating 1s and 0s to suggest that its subject matter is binary-encoded (digital) information. The binary pattern itself, though, is too perfect to contain meaning: an endless unvarying stream of 10101010 expresses nothing, except perhaps the presence of an idle or unusable electronic device. A key principle of information theory is that "[n]either noise nor information is predictable" (Kurzweil 30); meaning, even when symbolized by binary digits (bits), must be complex and unpredictable. On the cover of *Race in Cyberspace*, the perfectly predictable binary overlay is simply decorative, which may in itself be meaningful in terms of identifying the book's true subject matter: like the *Hypertext* series, *Race in Cyberspace* is about people, not computers.

Hypothesizing Versions

Paper-like texts, with their versions identified as "Revised" or "2nd edition," increasingly reside in digital libraries and personal collections alongside software-like texts, with their versions identified as "2.7.1"; caretakers of such collections may especially benefit from the development of version-comparison tools applicable to all kinds of texts. For digital texts as for others, the evolution of change through multiple versions can be usefully tracked, providing insight not only into the progress of the text itself but into the development of the technologies by which it was produced. To do this, possible sources of change must be identified so that the differences they can create are recognized: just as awareness of the process of stop-press corrections can explain differences between copies of the "same" printed text, so can understanding the process of activating a new blog template explain how the "same" blog posting can have a drastically different appearance from one minute to the next. Identifying the possible and the normative transformations among multiple textual containers can usefully suggest a text's place in its evolutionary context. For example, manuscript is usually a source for print; the reverse transformation, creating a hand-written text from a printed source, is possible but unlikely outside of unpublished projects such as diaries and academic notes. Similar recognition of possible versus likely transformations involving digital documents makes it possible to hypothesize about the relative positions of multiple versions in the history of a work, although additional information may be required as supporting evidence. For example, a Web page heavily decorated with images can be saved as a text-only version of itself in one operation (File—Save Page As—Text Files in the Mozilla Firefox browser); that transformation cannot be reversed automatically. A text-only Web page that refers to non-existent images is probably either an early pre-publication version, with its linguistic content in place before the decorative elements are added, or a late special-purpose version, pared down to plain text for efficient reading via a slow network connection or a small screen; identifying the true cause of differences from among the many possible explanations can require investigation into cultural practices that exist beyond the text.

For software-like hypertexts such as blogs, "powered by" multiple layers of independently-changing software and design, versions are appropriately described in multiple layers that separate content from its containers: changed content is identified by a timestamp; the structure-creating software and the design-creating theme are separately identified by their own names and version numbers. For paper-like texts such as books, a multi-layered approach to versioning can also be developed; attending separately to changes in content and changes in structure can produce, as it does for software-like texts, an appropriately granular understanding of the ways in which intentional changes shape an evolving text.

CHAPTER IV

"VIEW PAGE SOURCE"—TEXT, TOOLS, CODE, AND METACODE

Software-like hypertexts differ from other kinds of texts in that their sub-surface ultrastructures control the surface-level messages and behaviors available to human readers. In the preceding chapters, I identified some of the ways in which hypertexts' multiple layers of structure, and the separations between container and contents enforced by that structure, can be used to explain accidental or intentional change in two large categories of digital texts: software-like texts such as most Web pages, with direct access to deep structural layers; paper-like texts such as electronic paper, with direct access only to surface-level structures such as Tables of Contents. In this chapter, my focus is only on software-like texts such as Web pages. For such texts, I drill down into ultrastructure using extended examples from two perspectives on the creation and maintenance of digital hypertexts: that of the designer, who constructs a Web page, and of the webmaster, who acts as its caretaker.

First, by creating a simple HTML page and demonstrating internally-different methods of producing what appears to be the same result in that page, I argue for an understanding of hypertext as not only multi-level but multi-part, assembled of prefabricated components in a way that is not essentially different from the way in which moveable type is assembled into forms for a printing press; this is a Web page designer's perspective on hypertext, explored in the "What the Designer Saw" section of this chapter. By making choices among internally-different methods of creating the "same" external result in a hypertext, the designer can express complex ideas about the place of power in relationships between people and technology: Is it more important to use computer resources efficiently or to respond to human choices flexibly? I argue that, beyond expressing skill and creativity just as the author of a paper text does, the designer of a hypertext can make internal structural choices to express ideas about relative priorities among textual components and between content and container: If an image is smaller than the space reserved for it, should the image be repeated to fill the space? If a paragraph is too large to fit readably in the space reserved for it, should the paragraph be truncated or should its font size be reduced? If words and images are of conflicting shapes but must be combined, which component must be deformed to create the combination? I further argue that, because software-like hypertexts can carry within themselves sub-surface instructions (metacode) directing the future activities of search engine spiders and Web browsers, as well as historical and explanatory content (comments, abandoned code, decision trails) providing guidance to future caretakers, these texts contain a *hyper-* power to direct their own futures and record their own pasts.

In the "What the Webmaster Saw" section of this chapter, I examine data describing a Web page's patterns of usage; such data is routinely available to a page's webmaster but not to its surface-level readers. By analyzing data collected by a software tracking device imbedded within a hypertext, a webmaster can use the hypertext as a bridge between its author's intentions and its reader's actions, providing an ongoing awareness of how the text is being used and an ability to bring the author and the collective reader into closer alignment over time. I argue that digital hypertexts' ability to automatically create texts about themselves, recording patterns of their readers' activities, is an essential difference from printed texts. In response to a Web page's description of its own use, a webmaster can create multiple layers of collective interactivity, learning from the behavior of past readers to adapt the page to more appropriately attract future readers: internally, metacode can be adjusted to deliver the page differently to search engines; externally, content can be changed to reduce the page's attractiveness to an undesired audience. Usage data obtained from my own practice as a webmaster supports my argument that the Web does not eliminate boundaries so much as it obscures them from surface-level observers: from the webmaster's perspective, readers' places within geographical and other boundaries—public versus private, commercial versus academic—are obvious.

Stage Directions: Watching Versus Reading the Play

It may be that, as blogmaker WordPress.org's motto states, "Code is Poetry"; the similarly-central roles of structure, pattern, and design in both software and poetry certainly argue for this. For the same reasons, it may be that hypertext, created by code, is drama; if so, much of the on-stage presentation of the drama is directed by markup languages such as HTML, SGML, and XML. Markup languages instruct Web browser software as to how the text of a Web page is to be performed (bold face, centered, red), just as stage directions instruct actors how to perform the script of a play (whispering, stage front, while twisting a handkerchief). If the reader of a Web page is not a Web browser, just as if the reader of a play is not an actor, it is still possible for the reader to understand the text without being distracted by the interwoven instructions, and in fact to incorporate an interpretation of the instructions into the experience of reading the text.

As with drama, it may be appropriate to use different terms to describe different kinds of engagement with a Web page: an audience member may *see* the play or page, while a scholar may *read* it. While watching a performance of a play, the instructions are not visible but are put into action, producing visible results; while reading a text of a play, only the instructions are visible and their results must be imagined. Figure 30 relates a simple script with stage directions (HTML) to a brief performance of the Web page it defines.

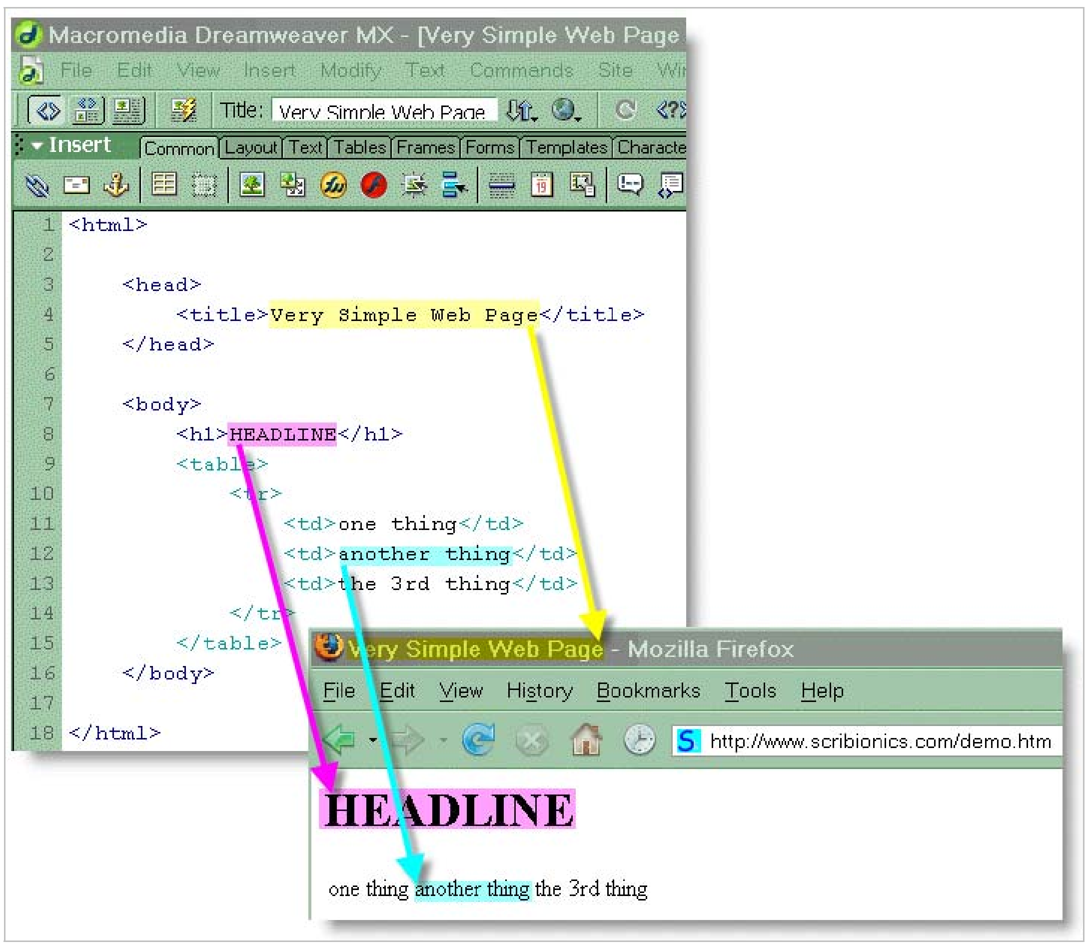


Figure 30: HTML source statements and the Web page they define.

This example uses Hypertext Markup Language (HTML), the most popular of the several markup languages in which hypertext pages can be defined, to create layers of containment and formatting into which text can be placed. All HTML pages share the basic structure demonstrated here. The page consists of everything between the <html> and </html> markers, shown in this example on Line 1 and Line 18. Bracketed between <html> and </html>, the page is divided into a head and a body. The head section is delimited by <head> and </head>, here on Line 3 and Line 5; the body consists of everything contained between <body> on Line 7 and </body> on Line 16.

The head section of a Web page contains backstage instructions, creating a context in which the page can be used; the code here communicates directly with search engines and Web browsers, and thus indirectly with human readers. In this example, the Firefox Web browser recognizes the text bracketed by <title> and </title> on Line 4 as the page's public identifier, and appropriately places that text ("Very Simple Web Page") above the page itself as the label of the browser window; <title> text is also customarily included as a footer if the page is printed, or in the page's description if it is listed as a hit by a search engine. The head section can include optional elements not shown here, such as self-descriptive metacode keyword lists and descriptions designed to help search engines appropriately retrieve the page in response to search requests relevant to its contents. Ultrastructures within the head section can also instruct Web browsers about behavior more complex than displaying a title line; for example, the <meta http-equiv="PRAGMA" content="NO-CACHE"> statement instructs the browser to treat this page as rapidly changing, re-loading the entire page at every visit so readers are provided the most current contents. The head section can also be used to send potential readers away: for example, <meta name=”robots” content=”noindex,nofollow”> tells any search engine that finds this page to ignore it, so any future searcher cannot use that search engine to find the page.

Operating within the environment defined by the head section, the body section contains the equivalent of on-stage instructions: text and its formatting (speeches); placement of images (scenery); performance of complex behaviors (controlled by small programs called “scripts”); hyperlinks to related pages (transitions to other scenes). In the very simple example of Figure 30, the body requires only the performance of a few words: on Line 8, <h1> and </h1> bracket a word to be shouted loudly (“HEADLINE”); between Line 9 and Line 15, <table> and </table> bracket words that need not be shouted but must be spoken in the specified order, as three data cells within one row of a table. While a text-only performance of this kind demonstrates some of the basic kinds of containment within a Web page (“another thing” is contained within a cell in a row in a table in the body of the page), most modern Web pages and other hypertexts are constructed of more complex components which, unlike a simple message such as “another thing,” must exist as separate entities before they can be assembled to create the page.

What the Designer Saw: Three Ways to Build a Blue Box

Like printed texts, digital texts are constructed of lower-level components. The nature of those components is not apparent at the surface level of the text; however, for most HTML pages and other instances of what I have called software-like texts, sub-surface levels are readily accessible. Figure 31 illustrates the surface-level appearance of a fragment of a Web page at www.scribionics.com/demo-boxes.htm, in which three blue boxes look identical but are constructed differently.

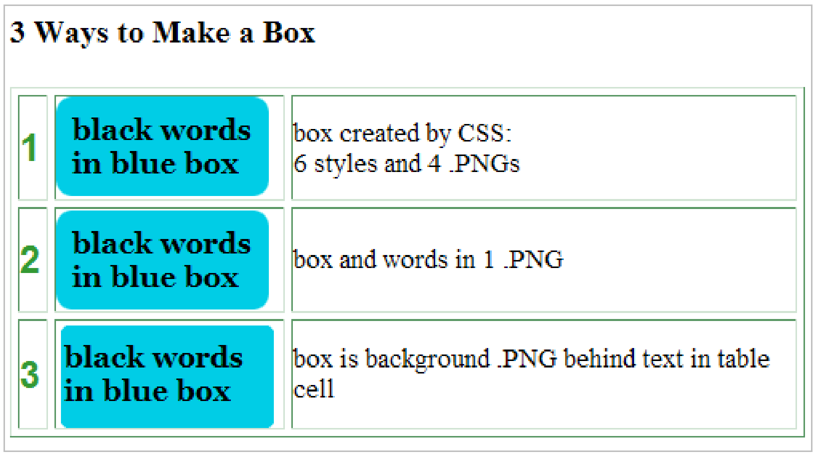


Figure 31: Three rows of a table look similar but differ internally.

In this simple example, different approaches to constructing the same text ("black words in a blue box") create differences that affect sub-surface concerns such as how quickly the page loads in a browser, how easily it can be changed, how its offline and online appearances differ, and how directly it can be transported into other digital texts. The central portion of the ultrastructure that created the three-row table in Figure 31 is shown in Figure 32.

```html
<table width="49%" border="1" cellspacing="4" cellpadding="0">
    <tr>
        <td width="4%" class="rownumber">1</td>
        <td width="18%"><div class="bl"><div class="br"><div class="tl"><div class="tr"><br>
        <span class="description">black words in blue box</span></div></div></div></div><div class="clear"> </div></td>
        <td width="78%">box created by CSS: <br> 6 styles and 4 .PNGs</td>
    </tr>
    <tr>
        <td class="rownumber">2</td>
        <td><img src="images/words-in-box.png" width="133" height="64"></td>
        <td>box and words in 1 .PNG</td>
    </tr>
    <tr>
        <td class="rownumber">3</td>
        <td height="69" background="images/wordless-box.png"><br>
        <span class="description">  black words <br> in blue box</span></td>
        <td>box is background .PNG behind text in table cell</td>
    </tr>
</table>
```

Figure 32: Ultrastructural differences affect matters other than the appearance of the table.

The definition of the table begins with a statement of its size and appearance (<table width="49%" border="1" cellspacing="4" cellpadding="0">) and ends with a matching closure (</table>). Within the table, rows are delimited by <tr> and </tr>; within each row, data cells are bounded by <td> and </td>. While the shape of the table is created within the <table> structure, the content of the table can be obtained elsewhere; in this example, all three rows depend to some extent on components found beyond the definition of the table itself.

Of the three methods used in this example, the HTML that defines Row 1 is the most complex to create and the most flexible to use, with the blue box dynamically sized to fit around the black text; in reality, there is no blue box here, just text on a colored background, with externally-stored images of rounded corners placed to create the illusion of a box with rounded edges. The method in Row 2 is simple to use but completely inflexible: the text is an integral part of an external image of a blue box, and neither the words nor the color or size of the box can be changed by HTML. In Row 3, the box is also an external image but the “black words in a blue box” message is specified and formatted by HTML; changing the HTML to place different words in the same space would still create the appearance of words in a box, although the box would not be resized if the words required additional space to appear “within” it. Creating black text in a blue box by any of these methods requires components beyond the <table> definition: Figure 33 shows all the images and styles used in creating this three-row table. The colored background of Figure 33 highlights one normally-hidden truth: on the Web, all images are rectangular, with the illusion of curves created by changes in color within a rectangle.

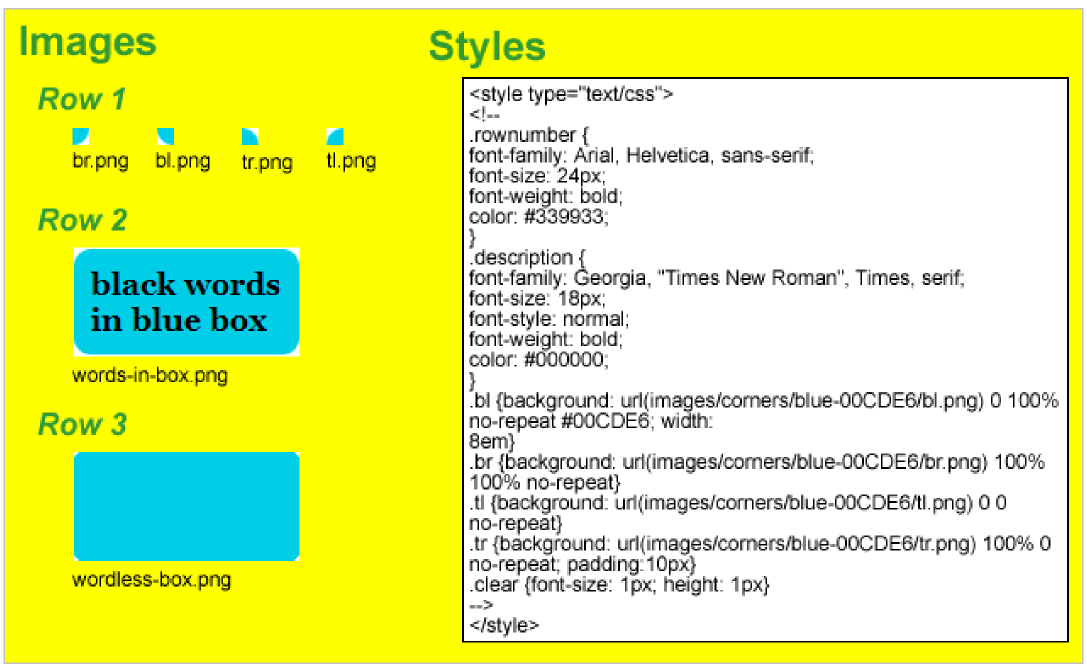


Figure 33: Parts list for the 3-row table includes 6 images and 7 styles.

The three-row table of blue boxes shown in Figure 31 is constructed of messages, images, styles, fonts, colors, and the HTML defining the table and arranging the other components within it. For Row 1, the blue box, its corner pieces, and its message are formatted by styles defined in the <head> and invoked in the <body>. Row 2 requires only one external component, an image combining the message and the box. For Row 3, a message formatted by a style is placed over an image of a box. Some of these components can be defined within the Web page before they are used; others must be created and stored outside the page. For all the components used in this example, Table 7 points out possible locations.

Table 7: Components used in a Web page can exist elsewhere.

	LOCAL: **on user's workstation**	**GLOBAL:** **on Web, inside current Web page**	**GLOBAL:** **on Web, outside current Web page**
messages		✓	✓
images		✓	✓
styles		✓	✓
fonts	✓		
colors	✓		
table		✓	

Messages can be defined within the Web page, as they are in Row 1 and Row 3 of this example. They can also be acquired from an external source such as a database, personalizing the page to greet a logged-in visitor by name, using the visitor's preferred language and displaying the visitor's current account balance or shopping cart contents;

in Chapter V, message handling of this type is the subject of a case study of collaboration and personalization in an open-source shopping site.

Images must be acquired from outside the Web page; in the example, the three methods of building a blue box use images stored in www.scribionics.com/images/, in the same site as the demo-boxes.htm example. The images were created similarly, as Macromedia Fireworks drawings stored in Portable Network Graphics (.png) format; the round-corner images are used within background styles, words-in-box.png is used as a foreground image, and wordless-box.png is used as a background image. Nothing intrinsic to these images requires them to be used differently within a Web page; the differences are created by the language that places the pre-existing images within the page. In addition to creating the surface-level appearance of a Web page, a designer's choice among the many possibilities for placing images on the page determines structural realities such as whether sizes are adjustable and how conflicts will be settled. In Row 1 of the table, <div class="br"> sets aside space to be controlled by the br style; in the stylesheet, .br {background: url(images/corners/blue-00CDE6/br.png) 100% 100% no-repeat}) places the bottom right corner image within any space controlled by that style, no matter what else might also occupy that space. In Row 2, <img src="images/words-in-box.png" width="133" height="64"> identifies a foreground image and reserves space for it; if the image is unavailable, the Web browser will display the empty space reserved for it; if the true dimensions of the image differ from the space reserved, the Web browser will distort the image to fit the space. In Row 3, <td height="69" background="images/wordless-box.png"> identifies the height of a data cell and the name of a background image to be placed within the cell; the width of the image becomes the minimum width of the cell; if the cell is wider than the image, the image will be repeated within the cell to fill it.

Unlike images, *styles* can exist within the Web page in which they are used; to maintain consistency among related Web pages within a site, styles are often defined in a separate stylesheet file that is pointed to by all the pages that use its styles. Some elements that appear to be images (colored backgrounds, for example) are actually styles that, like the .br style mentioned above, use images.

A designer's preferences of *fonts* and *colors* are specified within a Web page but those specifications can only be obeyed if they are within the capabilities of the end user's computer. Fonts are installed on each user's personal computer; if the specified font does not exist on a computer, the Web browser will select an alternate; in the current example, the .description style specifies that if the Georgia font is missing, the Times New Roman font is an acceptable substitute. Similarly, because software can define some colors that cannot be produced by some hardware, a Web design can specify a color that a user cannot see; in such a case, the user's computer will approximate the color as best it can.

While content such as messages, images, and styles can be acquired from sources external to the Web page, HTML-created containers such as <table> are defined within the Web page. Other common Web page components such as hyperlinks and scripts are not used in this simple Web page. *Hyperlinks* have been much studied as the defining attribute of hypertexts and the enabling technology of the Web; my project here is to argue that hypertexts are created of and defined by many other components, several of which are identified above, most of which must be defined before they can be used in a page. A simple *script*, acting as a tracking device to record the activities of visitors to a Web page, creates the data that is the focus of the "What the Webmaster Saw" section later in this chapter.

Alterity: Actively Seeking Internal Variation

In human languages, there are often multiple valid methods to communicate an idea; the differences among those methods are expressive of personal style and creativity, but also of social class, educational level, and other cultural elements. Similarly, there are multiple valid ways to create the same appearance in a software-like text. In *Cyberculture*, Pierre Lévy identifies "an openness toward alterity" as one of the essential values of "[t]he social and cultural movement that underlies cyberspace" (112). I argue that, as shown in the preceding examples of simple Web pages, the truth of this for hypertextuality goes much deeper than a passive "openness": in the construction of hypertexts, not only are alternate approaches to the same task possible and welcome, but they are actively developed as methods of expressing sub-surface ideas. Some of these hypertext-shaping ideas are about the appropriate place of power in relationships among words, images, and space; some are based on expectations of instability and the contributions of collaborators; some enable or prevent creation of alternate versions of the text.

To some extent, choices among coding techniques are motivated by personal preference and environmental factors, just as word choice can be. On another level, because computing languages are used with some expectations—ease of maintenance, for example—that are not ordinarily attached to human languages, the ramifications of choices go beyond whether their results are satisfactory in their current container. Choosing one of several valid ways to produce a result can mean that, while it operates as intended in its current container, it will not be portable into a future container. Web browsers such as Mozilla Firefox routinely offer multiple methods of copying all or part of a Web page into a new textual container. One method, the File—Print tool, freezes all the current instantiations—a style applied to a message, an image placed within a table—into a snapshot of the page's current surface-level contents. Even if the snapshot is stored as electronic paper it is a *still* image, no longer potentially variable or responsive to its environment; loss of access to a hypertext's sub-surface components creates a text that is flat, paper-like, not *hyper*-. Another method, using the browser's File—Save As Web Page tool, creates copies of all the components it can reach. The copy created is software-like, still *hyper*-, but is possibly incomplete. For example, if the page includes data extracted from a database, only the current instantiation will be saved; the database itself is out of reach. The two-step Edit—Copy, Edit—Paste procedure acquires copies of what it can see, but it cannot include invisible components such as stylesheets; a copy created this way preserves surface-level content but loses ultrastructural features that control it. Figure 34 demonstrates the result of such an operation, copying the three-row table discussed above from its Web page into Microsoft's word-processing software.

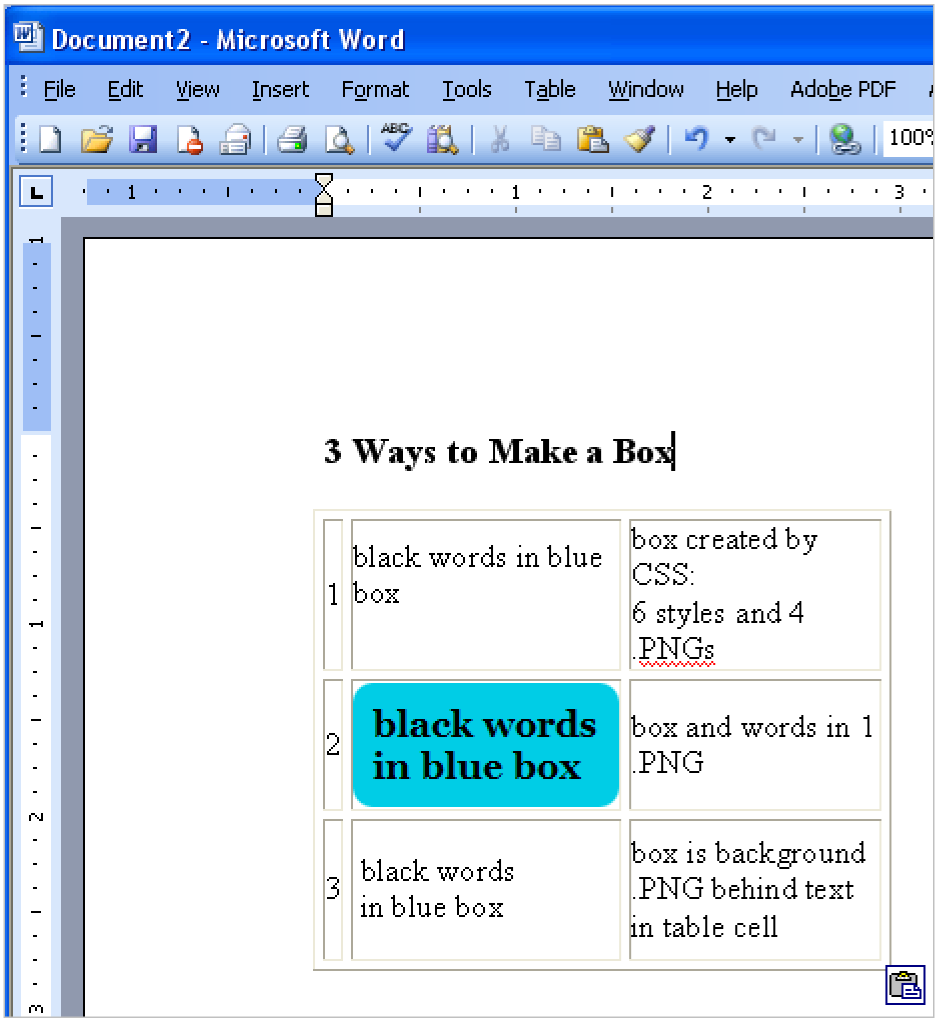


Figure 34: Pasting a Web page into a Word page causes loss of sub-surface components.

Here, the foreground image used in Row 2 is available at the surface and is copied successfully; background images and styles are cannot be copied, creating a defective copy in Word.

Namespace: Exposure, Disambiguation, Containment, Contextuality

Loss of internal components is not the only possible cause of damage when a hypertext is transported to a new container. An ultrastructure that validly shapes text in one digital container can be invalid or meaningless in another container; if the text is transported from one container into another in an inappropriate way, elements of the container's normally-hidden internal framework can be exposed to view as if they were part of the text's contents. For example, a completely healthy-looking digital text (a Web page, shown at the top of Figure 35) becomes, after passing through several layers of forwarding through electronic mail distribution lists, a digital text in which special characters are mis-handled and elements of the text's ultrastructure (xml:namespace …), meant to operate invisibly as computer-readable definitions of the text's container, are exposed as if they were part of the human-readable contents of the text: The difference between the clarity of the "AAPO First Friday Lecture" Web page at the top of Figure 35 and the confusion of the same message after transformation into an electronic mail message demonstrates that something beyond digitally-precise reproduction is required for correct communication: the container in which the message is received must be able to interpret its internal mechanisms in a way that is true to the design intentions of their author.

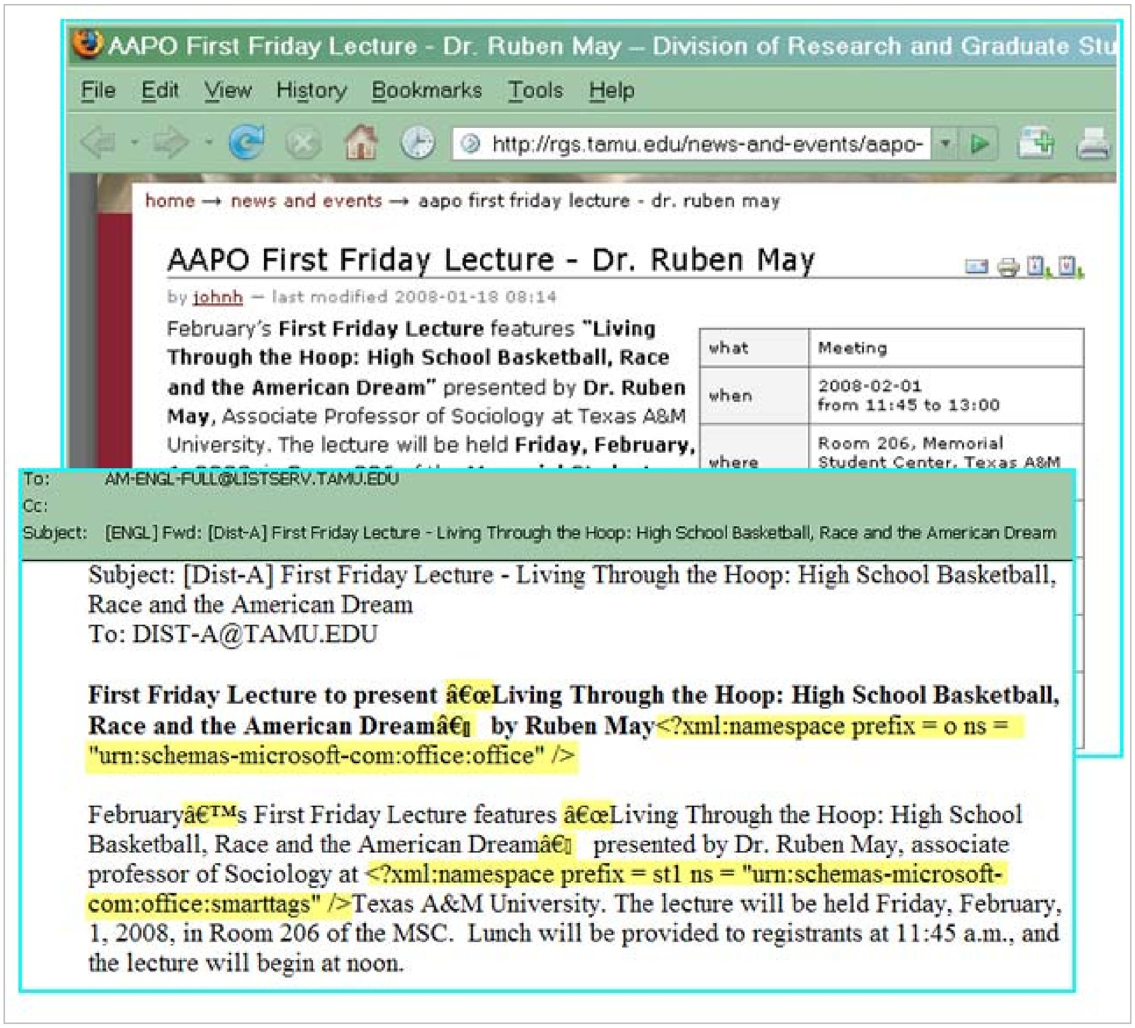


Figure 35: Properly-formatted iCalendar Web page (top) becomes improperly-formatted electronic mail.

Rather than being published digitally, if a campus-wide announcement such as shown in Figure 35 had been circulated on paper through several layers of distribution lists, the multi-generational paper copies might include other evidence of their distribution process: smudges and streaks from poorly-maintained photocopying equipment; misalignments and truncations due to carelessness by the persons who create photocopies; staple holes, dog ears, folds, drops of spilled coffee (and their images, preserved in photocopies); stray marks and comments added by previous readers. The difference is that, although all these errors accumulated in transmission are accidental, unrelated to the intention of the document's author, the errors introduced as a digital text is transmitted are errors of *exposure*: the problem is not that elements such as "<?xml:namespace prefix = o ns = "urn:schemas-microsoft-com:office:office" />" do or do not exist within a digital text, but that they are inappropriately presented at the surface level of the text.

A paper text certainly has a molecular structure, invisible to the human reader but serving to create the paper and the marks on the paper; disruption of that molecular structure—through exposing the paper to fire, for example—can cause loss of the message but can't cause molecules of paper to be mistaken for words of text. For digital texts, because their internal structure is also constructed of words, this is exactly what can happen: words meant as a message to a human can be intermingled with words meant to tell a computer how to convey the message. That is what happened as this announcement travelled through multiple levels of e-mail distribution: the message, created with Microsoft Office as indicated by its inappropriately-exposed XML, was added to the campus calendar system as an iCalendar (.ics) page, and emailed in HTML form (.htm) to the recipients of the "DIST-A" email distribution list; one of the recipients of "DIST-A" then passed it on to the "AM-ENGL-FULL" list, of which I am a recipient. No one involved in transmitting the message intended to expose ultrastructural material that is normally unseen, nor indeed to do anything with the message other than read it and pass it along; the message should have been preserved undamaged, in its original digital precision. The message content *was* perfectly preserved, but portions of its ultrastructure were imperfectly understood in context as it reached new containers.

As discussed in the *Google News* case study on containment error in Chapter II, computers' inability to relate text to context is at the root of difficulties in properly recognizing human cultural complexities such as humor. As shown Figure 35, computers can also have difficulty making sense of contextual information originated by other computers: context changes meaning, even among computers. To address this difficulty, in several computing languages (as in the XML fragment quoted above), and in other naming systems that rely on context to establish precise definitions, the concept of a *namespace* is central to successful communication: definitions "don't just float around in the code of your application" (Patrick 81) but are contained within a tree-like hierarchy, disambiguating them from similarly-named but differently-used containers elsewhere in the hierarchy. What begins as a meaningful contribution to the hypertextual infrastructure becomes noise when it is separated from its context, exposed as if it were the text of an electronic mail message.

While for printed texts the deep nature of the components of which they are made—ink, paper, binding—are studied within the realm of physical science, the components that make up digital texts are within the realm of information science. Because the languages humans use to communicate with computers strongly resemble natural human languages, this is important good news for textual scholars who wish to study the complete structure of digital texts; while it may take a chemistry degree to appreciate the nuances of ink's action on paper, training in human languages and literatures provides an excellent foundation for detailed examination of how hypertexts are built and how their structures create and limit their functions. In this, I agree with Mirielle Rosello that "I am not sure why hypertext should be put under the category of technology (as opposed to literature for example)" (123). Hypertextuality is one of many ways words can be used to express ideas; one key way in which hypertexts differ from other texts is that there are multiple layers of words within the hypertext, with the layers beneath the surface instructing computers as to the form and function of the surface-level words presented to the human reader.

Stray Marks: Meaninglessness in the Ultrastructures of Hypertexts

Inappropriately exposed on the surface of a hypertext, ultrastructural components such as the xml:namespace definition described above create errors in the visible content of the page: those words should not be readable in that context. For a paper book, errors in the *content* do not prevent it from being recognizable and usable as a book: it may be filled with false statements, but it is still properly sized and shaped and still operates as a book. Errors in the *construction* of a paper book, though, can be fatal to it: if pages are bound out of order, or the glue won't hold the pages, or the ink soaks through the paper, the book cannot be read and is perhaps not a book at all but a pile of useless scrap, awaiting recycling. For digital texts, errors in human-oriented *content* are likely to be just as innocuous as in printed texts: the words may be silly but they are perfectly readable, and the hyperlinks may connect to irrelevant pages but the connections are successfully established.

Web pages can also carry harmless errors in their ultrastructures, useless stray marks that have no effect on the appearance or the function of the page. Web design tools often leave abandoned formatting instructions in place; I call this unused markup “decision trails.” Although decision trails do not influence the design of the page as experienced by human readers, they can be useful in understanding the processes of development and experimentation that led to that design. For example, the formatting instruction WARNING has the same effect, from the human reader’s point of view, as the more complex WARNING. Where the first example expresses a straightforward decision that the word “WARNING” should be presented in a large, red font, the second example records three separate thoughts about it: it should be large; it should be blue; no, it should be red. Because the “red” decision is closest to the message, it overrides the “blue” decision: <font color=”blue”> has no visible effect on the published page.

The existence of decision trails, in addition to providing some insight into the designer’s thought processes, are strong indications of which technique the designer used in creating the page: the more obsolete code remains in place on the page, the more likely the human designer relied on WYSIWYG (“what you see is what you get”) design software, working at the level of creating a user experience rather than at the level of coding instructions to a computer. HTML coded without the aid of such tools does not typically carry visible decision trails: code-generating software is undisturbed by useless code, while human authors are distracted by it and are likely to immediately replace old code with new code rather than allowing abandoned instructions to remain in place.

In a hand-coded page, when abandoned code is potentially helpful in understanding a historical context, it may be retained as a comment, clearly marked as non-executable and wrapped with additional human-language explanation of the change. Comments are meaningful to the human designer or programmer but are not displayed to the human user of the page; the existence and thoroughness of comments within any code is another indicator of the extent of direct human involvement in the details of its creation and maintenance. Comments also provide insight into the human organizational processes involved in managing a Web page: an HTML comment such as `<!-- Wording and formatting of this warning are specified by the Safety Department; ask them before changing anything about it -->` indicates not only that a human designer rather than an automated design tool created the surrounding code, but that there is an expectation that future changes will also be manual and controlled by business, rather than technical, priorities. A Web page's ultrastructure can carry within itself its own history, such as abandoned and overridden formatting instructions; it can also participate in shaping its own future by carrying explanations and suggestions from one generation of its designers and caretakers to the next.

What the Webmaster Saw: Three Things to Do with a Copy

One difference between information distributed on paper and information distributed via the Web is that it is possible, with the help of the proper tools, for the Web publisher to maintain detailed, ongoing awareness of whether and how the information is being used. For a paper publication, minimal feedback is available about its success; most of that feedback is financial, based on whether the book or magazine or newspaper is being purchased, but financial data provides no insight into whether or how, once purchased, the text is being used: Is it purely decorative, unopened and beautiful on a coffee table? Has it been read and re-read, passed around and enjoyed and debated among a group of friends? Has a favorite page been ripped out, highlighted, and tacked up to a workplace bulletin board? Has a child snipped out the illustrations, glued them into a collage, and submitted the new creation as homework? What little post-sale information of this type is available to the author of a printed text arrives due to some extraordinary effort such as a published review, or a direct letter of praise or complaint; other than opinions expressed by those rare individuals who will take the trouble to actively announce their reactions, the author has little to go on.

In contrast, for a Web-published text a great deal of information is readily available describing its use, independently of financial transactions and any individual's effort to contact and inform the author. This information is available to the webmaster through software tools that, for every visitor to every Web page in the site, collect and analyze basic data such as geographic location, Internet Protocol (IP) address, and referring link. Awareness of visitors' geographic locations can motivate revisions of a site, such as translating informational pages into Spanish or adding a currency converter to an online catalog. Identifying the path each visitor follows to reach a Web page also provides key insight into how the page is used: Did the visitor follow a link from a forum or an email message, indicating that this Web page has become a topic of discussion? Was this page suggested by a search engine in response to a keyword search, and do those keywords relate to an unexpected interest in the subject matter, indicating a previously-unimagined audience who might benefit from additional information designed for them? Was this page opened directly, without the intermediary of a link or a search engine, meaning that the visitor knew the page's address through some offline means such as a printed advertisement, in which case it might be worthwhile to direct more of the advertising budget toward offline publication?

In my own practice as a webmaster I monitor and respond to usage information collected with several widely-used software tools, of which the most comprehensive and flexible is StatCounter, from the company of the same name at www.statcounter.com. StatCounter collects tracking information—not the visitor's name, but the name and location of their computer, with other details about how this visitor reached this page—for pages in which the webmaster has placed a tracking device: a tiny Javascript program, invisible unless the visitor examines the page's source code. As will be shown below, the imbedded tracking device also records the activities of visitors who pull a page out of the container in which they found it, not just reading the page but keeping their own copy.

Case Study: Tracking Unintended Uses of a Web Page

Several of the Web sites I manage are owned by a biotechnology company, Elsom Research Co., Inc.; of the hundreds of Web pages at the company's sites, one page is particularly problematic and thus will serve to demonstrate how a webmaster can identify and respond to issues of Web page containment. All the Web pages owned by

Elsom Research carry the StatCounter tracking device; using the StatCounter data, I was able to interpret and respond to a pattern of spikes in the site's popularity, at times so high as to exceed its bandwidth quota.

Focus: Unexpected Readers

A site which has exceeded its monthly bandwidth quota is like a wireless phone which has exceeded its monthly quota of usage minutes: such a site is dead, inaccessible to visitors until additional bandwidth is made available, either by purchasing more from the site's hosting provider or by waiting for the passage of time to reset the quota at the beginning of the next month. This is an important way in which Web-based publication differs from paper-based publication: if a paper book becomes popular, more copies may be sold to satisfy the demand, with the profits enriching the publisher; alternatively, if no more copies are produced, the increased popularity can increase the resale value of existing copies, enriching their owners. If a Web site becomes popular, the financial impact can move in the opposite direction, with the publisher required to purchase additional bandwidth to satisfy the demand or, by doing nothing, allow the site to go dead so that the increasingly-popular text is available to no one. This has come to be called the "Slashdot effect," after the self-described "news for nerds" site at www.slashdot.org which has repeatedly caused other sites to exceed their bandwidth quota by publishing words of praise about those sites: the result of Slashdot suggesting that a site is well worth seeing is that visitors flood the site until it is shut down and there is, perhaps briefly, no longer anything there to see.

The Slashdot effect is central to Christian Sandvig's argument in "The Structural Problems of the Internet for Cultural Policy": because bandwidth is a limited and therefore costly resource, only well-funded voices can be widely heard on the Web, so the Web is inherently anti-democratic. I appreciate Sandvig's project of identifying the design problems that cause technological advances to hinder rather than help social progress, but I think this observation is off the mark. Bandwidth does have a cost, but it is a low cost, becoming lower with the passage of time. For example, in October 2007 internet hosting provider Brainhosting charges $0.50 per month to add 1 gigabyte (one billion bytes) of bandwidth when a client site exceeds its initial 20-gigabyte quota; one byte is enough storage to represent one alphabetic character; fifty cents buys the resources to transfer one billion alphabetic characters from any publisher's Web site to any reader's Web browser. Fifty cents per month is within reach for even minimally-funded Web publishers; the Digital Divide is real, and it is certainly easier for some people (English speakers, especially) to speak online than it is for others, but the cost of bandwidth is not the root of that problem.

When, as in the case of Elsom Research, an organization's digital presence is its entire presence rather than an augmentation of some brick-and-mortar alternative, shutting down a Web site for exceeding a bandwidth quota is the same as shutting down the organization; access to the digital text is urgently required. As I will examine in detail here, the Elsom Research Web site was repeatedly shut down during the period of August-October 2007 due to exceeding its bandwidth quota. As the site's webmaster, I responded to each shutdown by increasing the bandwidth quota and returning the site to service, only to have the higher quota again exceeded and the site again shut down; it became essential to understand the source of the heavy traffic and, since there was no simultaneous increase in purchases by online shoppers, determine whether there was some way to prevent or redirect or otherwise adapt to it so the site would not require constant intervention to remain online.

I tracked the flood of visitors to one page within the site: "An Introduction to Nanosomes" at www.elsomresearch.com/learning/technology/nanosomes.htm. The page is a heavily-illustrated explanation of one of the company's anti-aging biotechnologies, consisting largely of drawings of cell membranes and their age-related changes. Most of the visitors to the nanosomes.htm page, which were most of the visitors to the elsomresearch.com site during the problematic period, had reached it in the same way: by searching the Web for illustrations of cell membranes. In addition to their interest in cell membranes, most of the page's visitors had something else in common: they appeared to be students or teachers, academic visitors rather than this commercial site's usual visitors from other companies or from shoppers. Figure 36 is a sample of the data collected by StatCounter identifying these academic visitors.

enm.collier.k12.fl.us (Collier County Public Schools) [Label IP Address]

Florida, Naples, United States, 0 returning visits

Date	Time	WebPage
22nd October 2007	09:59:11	images.google.com/imgres?imgurl=http://www.elsomresearch.com/images/nanosomir www.elsomresearch.com/learning/technology/nanosomes.htm
22nd October 2007	10:03:24	images.google.com/imgres?imgurl=http://www.elsomresearch.com/images/nanosomir www.elsomresearch.com/learning/technology/nanosomes.htm

vlanx5.cuw.edu (Concordia University) [Label IP Address]

Wisconsin, Mequon, United States, 0 returning visits

Date	Time	WebPage
22nd October 2007	00:08:25	images.google.com/imgres?imgurl=http://www.elsomresearch.com/images/nanosomir www.elsomresearch.com/learning/technology/nanosomes.htm
22nd October 2007	00:08:29	images.google.com/imgres?imgurl=http://www.elsomresearch.com/images/nanosomir www.elsomresearch.com/learning/technology/nanosomes.htm
22nd October 2007	10:01:04	images.google.com/imgres?imgurl=http://www.elsomresearch.com/images/nanosomir www.elsomresearch.com/learning/technology/nanosomes.htm

cd7f0804.nwservers.granite.k12.ut.us (Utah Educational Network) [Label IP Address]

Utah, Salt Lake City, United States, 0 returning visits

Date	Time	WebPage
22nd October 2007	09:41:03	images.google.com/imgres?imgurl=http://www.elsomresearch.com/images/nanosomir www.elsomresearch.com/learning/technology/nanosomes.htm
22nd October 2007	09:45:16	images.google.com/imgres?imgurl=http://www.elsomresearch.com/images/nanosomir www.elsomresearch.com/learning/technology/nanosomes.htm
22nd October 2007	09:45:56	images.google.com/imgres?imgurl=http://www.elsomresearch.com/images/nanosomir www.elsomresearch.com/learning/technology/nanosomes.htm
22nd October 2007	09:52:30	images.google.com/imgres?imgurl=http://www.elsomresearch.com/images/nanosomir www.elsomresearch.com/learning/technology/nanosomes.htm
22nd October 2007	09:52:44	images.google.com/imgres?imgurl=http://www.elsomresearch.com/images/nanosomir www.elsomresearch.com/learning/technology/nanosomes.htm

Figure 36: StatCounter data showing use of commercial Web page by academic visitors.

Like the other academic visitors during this period of heavy traffic in late 2007, these visitors used a Web search engine (in these examples, images.google.com) to search for images with keywords such as “cell membrane,” “cellular membrane,” and “cell membrane structure” and then, as shown in all three examples above, move repeatedly between the list of returned images (images.google.com/imgres?imgurl=…) and the nanosomes.htm page in which those images are displayed. Because images cause a Web page to rank highly in the list of hits returned by a search engine, and because nanosomes.htm has many images with keyword-matching names such as "membrane.gif" and a high percentage of "real" text (content-delivery words such as "membrane" appearing with great frequency relative to navigational words such as "click"), nanosomes.htm was consistently returned in the top positions for searches seeking illustrations of cell membrane structures.

The flood of visitors was caused by the creation of a successful Web page, if "success" is defined as "use": people were visiting the page. However, since success for a Web page is more correctly defined as use by the intended audience, the nanosomes.htm page was not successful at all; the page was designed to inform consumers about a commercial product and therefore motivate them to buy it, not to educate students in basic biology classes. Relating the Web page's subject matter (cellular structure) to the timing of its burst of popularity at the beginning of the Fall 2007 academic semester and similar (though lower-volume) peaks in visitor activity in January 2007 and September 2006 supported the interpretation of these visitors as participants in introductory biology classes: cellular structure is a key concept, introduced early in such classes; as the classes move ahead from studying single cells to studying multi-cellular plants and animals, the demand for bandwidth to support visits to nanosomin.htm should be expected to decrease, as it did.

By creating and publishing the text and illustrations and paying for the disk space and bandwidth to make and keep them available, Elsom Research has evidently been subsidizing educational efforts in introductory science classrooms world-wide for several years. Because I have usage statistics only for Web pages I control, I can only speculate about the extent to which this is happening to other enterprises whose “free” Web publications address other academically-interesting subject matter. Neither can I elaborate on the pedagogical effects of using commercially-motivated material as the basis of science instruction: if students’ earliest encounters with scientific argument are in the context of advertising, and if advertising claims are notoriously unrealistic, is a scientific explanation just one more anti-consumer trick? Preventing identifiably-academic visits to the nanosomes.htm page would address both the issue of bandwidth peaks at the beginnings of academic semesters and the concern about students learning science from text intended for shoppers. However, current technology does not provide the necessary granular control: a webmaster can password-protect a page so only pre-approved visitors can read it, or block visits by some or all search engine robots or spiders; for a page that is meant to be freely available to shoppers (who cannot be reasonably expected to request a password before pursuing their interest in a subject), there is no mechanism to filter out only an unwelcome segment of the human population.

While I could not make the nanosomes.htm page unavailable to academic visitors, what I could do was make it less appealing to them, as shown in Figure 37.

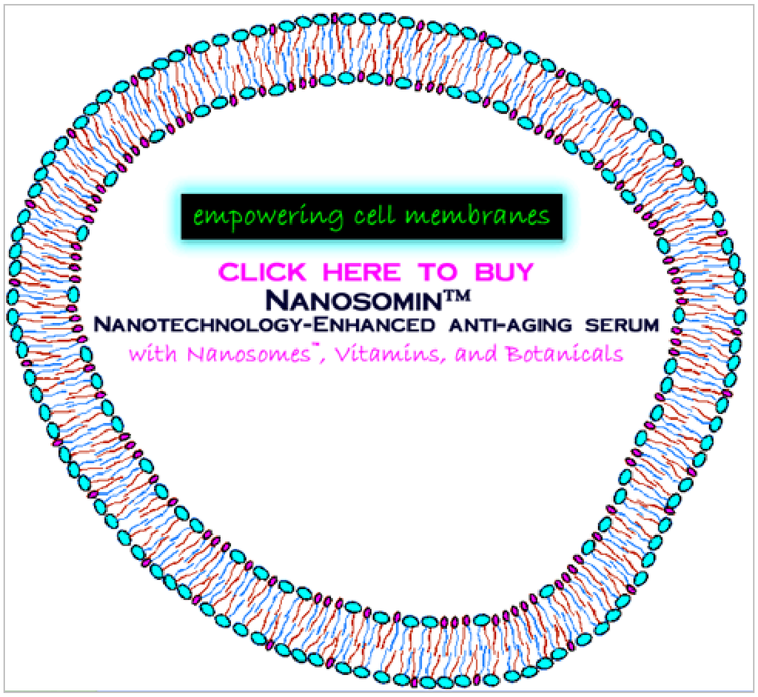


Figure 37: Disrupting illustration with commercial text reduces its attractiveness to non-commercial visitors.

By modifying several of the illustrations in nanosomes.htm to fill otherwise-empty space with commercial messages ("Click here to buy"), the page became less attractive as a text for direct classroom presentation or as a source of images to be copied and pasted into homework; while the blatant advertisement interfered with the page's usefulness to its unintended academic audience, it was not at all surprising or disruptive to the page's intended audience of potential shoppers. Commercial photographers and other visual artists routinely deform the images on their Web sites in a similar way, adding a watermark or other obvious defect so that the image on the Web is useful as a demonstration of the artist's talent but not as a no-cost alternative to the undamaged image, offered for sale.

The internal construction and multi-layered containment of a Web page is ordinarily *subtle*, from *sub+tela*, “under the cloth.” *Unsubtle* revision of the kind shown in Figure 37, pushing an obvious reminder of the nature of the Web page’s container (a commercial Web site, identified by its easily-ignored .com domain name) up into the surface layer of its contents, may be the most useful way to resist unwanted readership on the globally-accessible Web. This is the natural corollary of my argument in “Fighting Speech With Speech: David Duke, the Anti-Defamation League, Online Bookstores, and Hate Filters” (Coste): hostile *writing* on the Web can be more usefully resisted by writing about it than by engaging in endlessly-escalating attempts to block it. Here, I argue that hostile *reading* also occurs on the Web, not only in the accidental Slashdot effect of overloads by enthusiasts and in the intentional malice of denial-of-service attacks, but in the consumption of online resources by unexpected and unwelcome readers who use a Web page to support their own purposes (such as teaching about cell membranes) rather than those of the page’s author (such as selling a product). I further argue that unintentionally hostile reading, like hostile writing, is more effectively resisted by expanding the text (adding advertisements, in this case) than by attempting to hide it.

Focus: Real Geography

The ease of using texts for unintended purposes, ignoring the categories and containers in which their authors published them, is just one of many ways in which the Web’s behavior argues for an understanding of itself as independent of boundaries, a “distanceless home [...] extending to infinity” (Gibson 52). While the Web can provide its readers and writers with a sensation of communicating in “a pure no-place” (Nakamura 15) where national and personal boundaries are invisible and unimportant, webmasters have access to extensive data about the layered physical realities—a computer on a network in a school in a city in a country—in which Web visitors are contained. With this data as evidence, I argue that the user experience of the Web as “shape-shifting, borderless medium” (Saffo 17) is a surface-level illusion: the networking technology facilitates crossing of national boundaries, but below the surface those boundaries are nonetheless observable and real. Texts on the Web are stored in real places: for example, the nanosomes.htm page discussed above was written on a personal computer in San Antonio, Texas and is published via a Web server in Atlanta, Georgia; just as its author and publisher exist in identifiable locations, its readers can also be tracked to real places that can be named and located on terrestrial maps.

The geographic diversity of visitors to the nanosomes.htm page can be seen by examining a small sample of the data collected by StatCounter. For a 12-hour period near the end of the previously-discussed flood of academic visitors (midnight to noon Monday October 22, 2007), Table 8 locates identifiably-academic visitors to the nanosomes.htm page. Visitors are identified as academic based on the registered name of their IP address or, when no name is provided, by the name of the server at that address.

For example, the registered name “Concordia University” identifies a visitor using on-campus computing resources at that university; in the server name “server.catawba.k12.nc.us”, the “k12” portion of its address identifies it as associated with education at any level between kindergarten and twelfth grade inclusive.

Servers are conventionally labeled to identify their purpose and physically tied to a geographic location; students, however, can visit a Web site for an academic purpose with the help of computing resources intended for that purpose—a server at the school, for example—or the same student can visit the same Web site for the same purpose via computing resources—such as a public Wi-Fi connection in a neighborhood coffeeshop—not specifically related to the academic purpose. Visits from academic users who did not reach the site through an identifiably-academic Internet Service Provider (ISP)—students doing homework off campus, for example—cannot be recognized as such and are not included in this table. This points out one of the limitations created by relying on identification of a visitor’s computer as a means of identifying the nature of the visitor: humans routinely escape from their containers in a way that computers cannot.

Table 8: List of academic visitors to commercial Web page during one 12-hour period.

HOUR	IDENTIFIABLY-ACADEMIC VISITORS
0000	▪ Concordia University, Wisconsin **USA**
0100	*none*
0200	*none*
0300	*none*
0400	*none*
0500	▪ Taif University, **Saudi Arabia**
0600	▪ Universiteit Utrecht, **Netherlands**
0700	▪ Lowell Public School Department, Massachusetts **USA** ▪ Islip Free School District, New York **USA** ▪ Madison-Oneida Board Of Cooperative Educational Services, New York **USA**
0800	▪ Norwalk Public Schools, Connecticut **USA** ▪ University Of Shahrkord, **Iran** ▪ Capital Area Intermediate Unit, Pennsylvania **USA** ▪ server.catawba.k12.nc.us, North Carolina **USA**
0900	▪ Little Rock School District, Arkansas **USA** ▪ Utah Educational Network, Utah **USA** ▪ server.catawba.k12.nc.us, North Carolina **USA**
1000	▪ Concordia University, Wisconsin **USA** ▪ Collier County Public Schools, Florida **USA** ▪ Bloomington School District, Minnesota **USA** ▪ Springfield Public Schools, Oregon **USA** ▪ Upper Canada College, Ontario **Canada**
1100	▪ Montgomery County Intermediate Unit, Pennsylvania **USA** ▪ Columbus Public Schools, Ohio **USA** ▪ Bloomington School District, Minnesota **USA** ▪ Madison-Oneida Board Of Cooperative Educational Services, New York **USA** ▪ Los Angeles Unified School District, California **USA** ▪ server.catawba.k12.nc.us, North Carolina **USA** ▪ hamilton-gw2.metro06.phila.k12.pa.us, Pennsylvania **USA** ▪ Sioux Falls Public Schools, South Dakota **USA** ▪ Northeast Ohio Network For Education Technology, Ohio **USA**

Of the twenty-seven visits from academic sources during these hours, twenty-three are from schools within the United States; others originate in Canada, Saudi Arabia, Iran, and the Netherlands. The visitors from outside the US are all associated with universities; within the US, most of the visitors are from elementary and secondary schools. If the goal of the page were to provide maximum relevance to the majority of its readers, this information about the demographics of its readers would provide valuable guidance for future revisions of the page; for example, adding a suggested reading list of elementary biology texts might serve these readers well; commercializing the reading list, with hyperlinks to an online bookstore where they could be purchased, might serve the page's commercially-motivated publisher well. Since the goal of the page's publisher was not to enhance the experience of academic visitors to the page, the ready availability of user demographics created additional questions about the extent to which a publisher can reasonably control the uses to which Web publications are put.

Focus: Unauthorized Copies

When a paper copy is produced, whether by printing from a digital original or transcribing or photocopying a paper original, the keepers of the original can obtain little more than hints about the existence and uses of the copy. Reader behavior can leave some indications—reference books abandoned near a photocopier, for example—that a paper copy was produced, but the copy itself is silent. A digital copy of a Web page tracked by StatCounter, though, automatically provides ongoing updates on its activity, as illustrated in Figure 38.

VISITOR ANALYSIS	
Referring Link	No referring link
Host Name	
IP Address	195.43.5.33 [Label IP Address]
Country	Egypt
Region	-
City	-
ISP	National Research Center
Returning Visits	0
Visit Length	19 hours 33 mins 31 secs
VISITOR SYSTEM SPECS	
Browser	MSIE 6.0
Operating System	Windows XP
Resolution	800x600
Javascript	Enabled

Navigation Path

Date	Time	WebPage
30th January 2008	06:14:45	No referring link /E:/Fatma/NANO%20PPt/liposomes/Nanosomes,%20the%20Technology
30th January 2008	07:03:16	No referring link /E:/Fatma/NANO%20PPt/liposomes/Nanosomes,%20the%20Technology
31st January 2008	01:48:16	No referring link /E:/Fatma/NANO%20PPt/liposomes/Nanosomes,%20the%20Technology

Figure 38: Digital copy reports on its name and location.

This StatCounter visitor analysis report shows that "Fatma," through a server at the National Research Center in Egypt, acquired a copy of nanosomes.htm and stored it on a removable device ("E:") such as a thumb drive, CD, or diskette; she opened the copy three times in the course of two days, just as someone might do with a paper copy, carried in a briefcase and taken out to study at free moments. The "No referring link" indication in the StatCounter report shows that the page was opened directly, not by following a hyperlink from another Web page. Even when the page is opened without having been reached through the Web, the fact that it was opened is recorded in StatCounter's database if the computer reading the private copy had an active Web connection at the time; the Javascript embedded in the page sends a record of the page's use to StatCounter's online database, for later retrieval and analysis by the webmaster. "Fatma" opened the page all three times through the same IP address at the National Research Center; if she travels away from the National Research Center and opens it again, the IP address of her server will show her at a different location but the Web page in the same portable container she created, "/E:/Fatma/NANO…."

At the same time in late January 2008 that "Fatma" created and repeatedly examined her copy, two other readers created their own copies of the nanosomes.htm page. Figure 39 compares the activities of all three, showing that readers interact with digital copies in some of the same ways they handle printed copies; the difference is not in what the reader can do, but in what the webmaster can know about what the reader does.

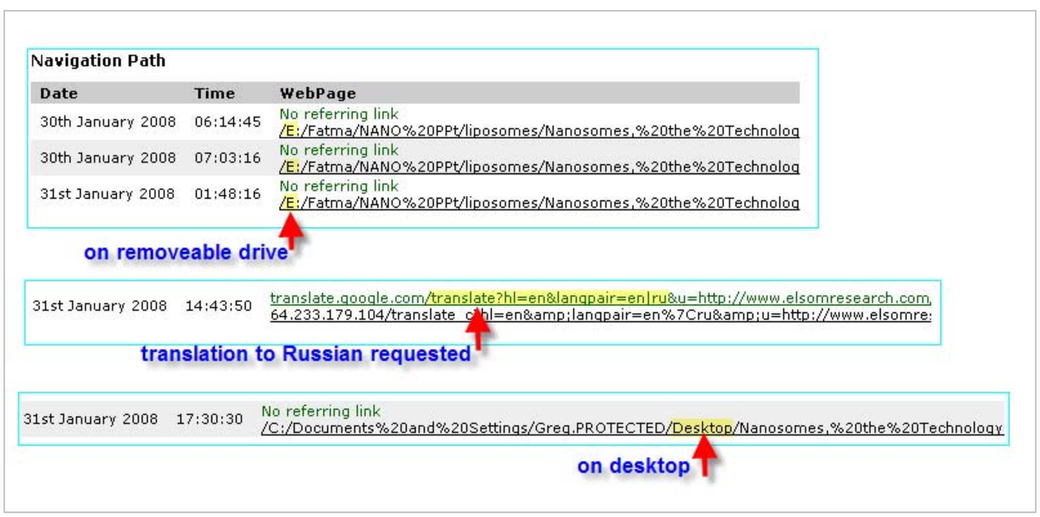


Figure 39: Evidence of portable copy, copy submitted to translator, and copy on desktop.

Just as “Fatma”’s placement of a digital copy on a portable device analogous to a briefcase is indicated by the file structure in which it is contained, so is “Greg”’s placement of a copy in another location typical of office users: his desktop. The “/C:/Documents and Settings/Greg.PROTECTED/Desktop/…” structure places its contents in the top level of the Windows user interface, where “Greg” will see it frequently. The Windows simulation of a desktop, like a physical desktop, is not meant to be archival storage for inactive material (although in both cases the desktop can be misused in that way); placing the copy on the desktop rather than at some lower level in the file system suggests that “Greg” (online via the servers at Sabre Sciences in Carlsbad, California) is currently engaged with and highly interested in it.

Like “Fatma” and “Greg,” another Web page reader did something that a reader of a paper page might do: the reader asked for help in understanding the page’s contents. In this example, a computerized translation system at translate.google.com (Google Translate Beta) was asked to translate the text from its original English into Russian. None of the observable information about this activity suggested a personal name, such as “Fatma” above or “Greg,” but a physical location was recorded: Staten Island, New York, USA, through the Earthlink, Inc. service which provides Web access to private homes. The results of this translation, shown in Figure 40, point out another area of difficulty in using computers to create texts: a human translator could easily have recognized that this Web page, as many others, includes some images with internal textual labels (for example, one side of a dividing line labeled “Water” and the other side labeled “Water-Seeking Region”); where a human translator might have provided a translation for the labels in images as well as for the words found outside images, the translation software has failed to do so.

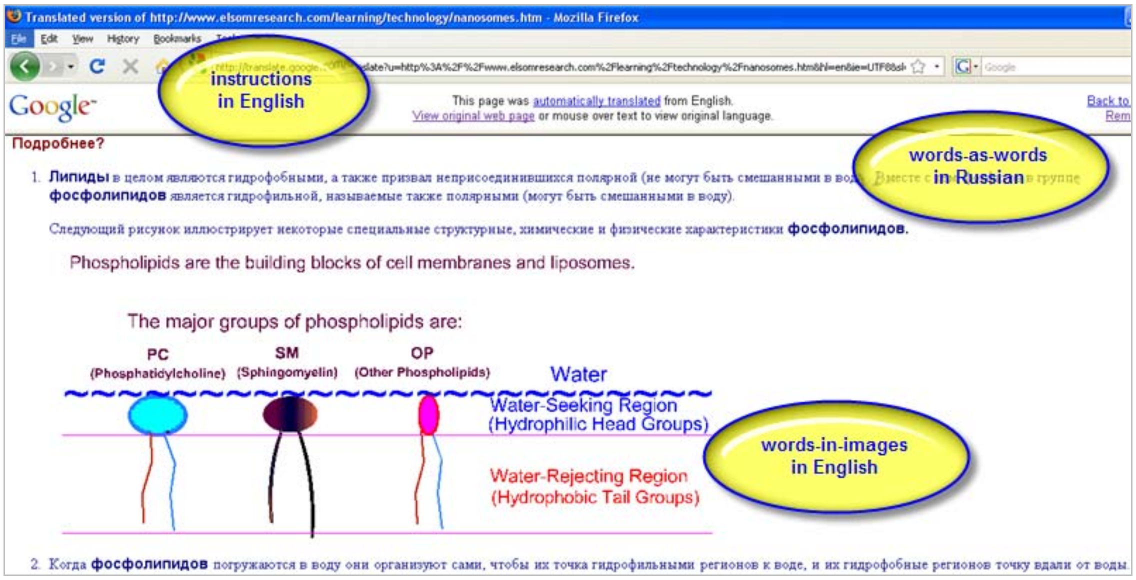


Figure 40: Machine translation of English page to Russian cannot translate words embedded in images.

For a human, all words are words and therefore available for translation into other words; for a computer, though, words-in-images are not words but images; images are not available for translation into words, so this machine translation creates a hybrid text in which words-as-words appear in Russian and words-in-images appear in English. This is neither what the author of the document created nor what the reader of the document hoped for in requesting a translation. It is however, representative of a broad category of text that is likely to resist accurate digital reproduction: text which must be available in multiple languages, especially when the boundary between words and images within that text has been transgressed.

Unlike the copies created by “Fatma” and “Greg,” the copy created by translate.google.com is ephemeral, created on demand by the translation tool; its address is a pointer to the translation tool, followed by list of parameters instructing the tool as to the location and desired handling of the source document. What might appear to be a hyperlink to the translated copy is actually instructions for creating the copy, as translate.google.com/translate?u=http%3A%2F%2Fwww.elsomresearch.com%2Flearning%2Ftechnology%2Fnanosomes.htm&hl=en&ie=UTF8&sl=en&tl=ru asks for the contents of nanosomes.htm to be translated from a source language of English (“sl=en") to a target language of Russian (“tl=ru"). Unless the human visitor takes steps to save the results, the translated copy exists only potentially, in the sense that executing the same instructions will produce the same input from the same output.

Tracking data reports the existence of a copy and some details about its location and use; every time the copy is opened, StatCounter reports additional data about its use, perhaps providing further insight into its life independent of its original but never directly answering questions about human behavior and motivation, such as why the user felt the need to keep a private copy rather than returning as needed to the public original. If the keeper of the private copy alters its internal structure to remove the StatCounter tracking device, there will be no further indication that the copy exists: the digital copy will become as silent as a paper copy. When contact is lost with a server, tools such as “ping” are used to search the network for it, just as a sonar operator can send signals through the ocean in search of a missing ship; no tool of this kind exists to allow the keeper of an original Web page to search for signs of previously-active copies. Tracking data such as that provided by StatCounter is strictly passive; there is no method of searching the Web for a copy that has gone silent.

Layered Perspectives: Reader, Writer, Designer, Publisher, Webmaster

A reader's experience of a Web page—what it looks like, how it behaves—can be so unlike the experience of a printed page that the surface-level differences—hyperlinking, interactivity, animation—seem to define and explain hypertextuality. A Web page developer's experience of the three-part process of writing a Web page—building containers, creating content, placing the contents appropriately within the containers—gives a multi-level experience of what hypertextuality is, based on the processes by which hypertexts are constructed. These separate paths to understanding the "same" text are as old as reading and writing; the details differ, but the sometimes-separate perspectives of readerly use and writerly creation are of long standing. A new and complex perspective on software-like hypertexts, unavailable for other kinds of text, is that of the webmaster. The webmaster's access to privileged information about a hypertext—who uses it, when and where, and for what purposes—creates opportunities not only to observe reader behavior but to respond to it. By altering a Web page's supporting environment—adding bandwidth, blocking spiders, redirecting hyperlinks, expanding keywords—a webmaster can change its patterns of use, making it available to different readers or for different purposes; this depth of post-publication awareness and control gives digital hypertexts a uniquely direct two-way connection between publisher and reader.

CHAPTER V

"EDIT THIS PAGE"—COLLABORATION AND CONTROL

Looking for examples of collaboration on the Web is like looking for examples of geology in a rock garden: at every level, from the chattiness of human threaded discussions to the silent packet-switching of network nodes, the Web is made of and used for cooperative communication, combining the efforts of multiple contributors into ever-larger texts. Many of the ways in which texts can be collaboratively produced are as applicable to paper-based media as to digital hypertexts, but I argue that the *hyper*-collaborative possibilities available in the construction of hypertexts are created by three interrelated elements that are not available to printed texts: multiple layers of control; multiple directions of movement; persistent access to historical and structural components.

The Web has no center, and therefore no central control; I argue that, choosing any single Web site as its own center, the strength of central control over textual collaboration weakens as distance from that center increases. The structure of any single site is likely to be tightly controlled by its webmaster; if the site's content supports multiple authors as in a forum or blog or wiki, there may be multiple controlling voices among those authors, managing aspects of the discussion with different levels of power as moderators or administrators; I illustrate this site-specific power structure in the "Layers of Control" section of this chapter. While each site's local controllers have power over the structure and content of the site itself, in the broader scope of the Web they become powerless to limit the ways in which the site participates in larger collaborative texts: anyone may establish their own hyperlink to the site, or aggregate it into the findings of their search engine or news reader. Additionally, positions of power in the collaborative authorship of a site can be granted and revoked as needed, which I argue is another unique feature of collaborative control in a hypertext: the publisher of a paper text can cease operation, but will always be on record as having been the publisher of that text on that date; the moderator of an online forum can be seamlessly replaced, immediately gaining control over past content as completely as over present and future content.

A related *hyper*-collaborative feature of hypertexts is the ability to move an entire collaborative production, not just some element of its contents, backward or forward in time. In a paper-like text, content can skip generations (material from Version 1 can be omitted from Version 2 and restored in Version 3) but the order of generations is unalterable. Full-volume reprints of old books are themselves unavoidably new: even if a 2009 reprint of a book first published in 1909 did not identify itself as such on its copyright page, the briefest examination of its paper and binding would reveal that, while the words might be original, the book itself is not; if the new reprint could not be readily distinguished from the century-old original, this confusion would be due to an intentional act of fraud rather than to any accidental blurring of lines between publication and re-publication. However, as I illustrate in the "Directions of Movement" section of this chapter, generational change in hypertexts need not proceed in any fixed order; today's version is today's version, not last week's version as re-published today. Word-processing software features such as Edit—Undo and Edit—Redo allow a solo author to move a privately-controlled digital text a few steps backward or forward in time; I argue that for some collaboratively-written hypertexts there are also *hyper*-Undo and *hyper*-Redo tools, acting on the combined work of all collaborating authors to move the entire production to a different point in time. I further argue that textual movement of this kind is profoundly different from reprinting old words (contents) on new paper (container): when a wiki page is relocated in time, all its components move simultaneously to the new location.

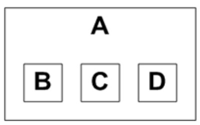
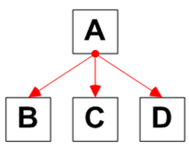
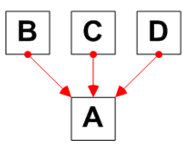
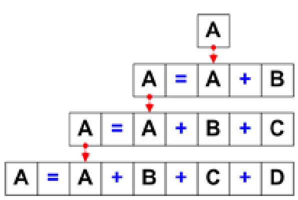
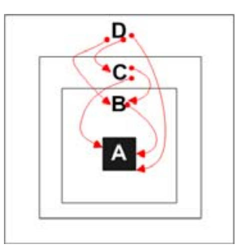
A third *hyper*-collaborative feature of many hypertexts, enabling and complicating the two features mentioned above, is the permanent identifiability (not always at the surface level) of all the components of the collaboration. Collaboratively-constructed hypertexts such as wikis, blogs, and forums store separate contributions, along with the identities of their contributors, within their databases. The Web is famously supportive of the anonymous and the ephemeral, but I argue that these are surface-level phenomena, obscuring the structural reality that the history of a hypertext's collaborative construction, including connections between authors and their words, even words they have retracted, can be recorded and retrieved. In the "Persistent Access to Components" section of this chapter, I illustrate a portion of a blog's database, showing that multiple revisions of the "same" posting are preserved as separate records; while only one version is published at any given time, all versions exist and, with the proper access to the hypertext's infrastructure, can be examined and modified.

Some Web sites are structured to allow visitors to choose the human language in which the site should speak; in the "The Narrator's Voice in Dynamic Web Pages" in this chapter, I expose the internal mechanisms of one such site. I argue that a *hyper*-narrator can be a component of a hypertext, serving a function similar to that of the narrator in a work of fiction even when the text itself is non-fiction; a *hyper*-narrator can be selected to meet the specifications of the visitor, then replaced by any of its alternative voices if the visitor's preference changes. For one dynamic Web page from the same site, I demonstrate a granular description of its components, identifying the extent to which each one contributes to the dynamism of the page; similarly, in the "Models of Collaborative Authorship" section immediately below, I offer visual models of some of the ways in which a text can be collaborative. I argue that studies of hypertextuality must begin to operate at this level of detail: not only identifying a hypertext as adapting to each reader, but pointing out the ways in which its components are assembled within its structure to make this adaptability possible; not only identifying a hypertext as collaborative, but describing in what ways it is collaborative.

Models of Collaborative Authorship: Fifteen Ways of Looking at a Hypertext

The nature of collaboration in a specific text can be described in a clear and granular way, just as the nature of accidental and intentional change can be described, and with similar benefits. In the three tables that follow, I have created and named fifteen visual models of collaborative authorship; each table is followed by short descriptions of the collaborative models listed. For most models, I have suggested examples of both digital and printed texts in which the model is implemented; if no example of the model is available, I have grayed out the box in which it would otherwise be listed. While these models differ from each other in some obvious, surface-level aspects of appearance and behavior, their more important differences are caused by different understandings of the places of power in relationships among collaborators and among the separately-contributed texts that are the components of the whole, collaborative text. Table 9 introduces the first group of models of collaboration.

Table 9: Five models of collaboration: aggregated, centrifugal, centripetal, collective, encapsulated.

model	name	digital example	print example
	**aggregated**	Really Simple Syndication (RSS) feed	anthology
	**centrifugal**	the Web	bibliography
	**centripetal**	viral marketing	advertising
	**collective**	wiki	1st, 2nd, 3rd, 4th editions of same title
	**encapsulated**		*Talmud*

An *aggregated* text is the product of many independent voices (here represented as B, C, and D) whose work is rolled together and presented as the product of another voice (A) which is the text presented to readers: access to B's text is mediated by A's text. In Chapter II, much of the discussion of unintentional change focuses on aggregated texts such as *Google News*.

Centrifugal and *centripetal* collaborative texts are constructed similarly, with one voice serving to direct readers to the work of another voice, but they direct traffic in opposite directions. A centrifugal text pushes traffic *away* from a center: a Web page that is the product of a single voice (A) directs readers to material created independently by many other voices, providing a direct route to that material via hyperlink; B need not cooperate with nor even be aware of A's connection. Moving in the opposite direction, a centripetal text pushes traffic *toward* a center: by publishing hyperlinks and recommendations, multiple voices direct traffic toward the work of a single voice, such as an advertiser's sales page; this model is applicable to search-engine optimization projects, viral marketing campaigns, and other multi-pronged efforts at increasing the visibility of a single page.

The *collective* model of collaboration is especially applicable to wikis, in which all readers are invited to write and all writing can be applied to the central text rather than being relegated to the supporting status of responses or comments. With a different approach to the central text, in *encapsulated* collaboration the contribution of the initial collaborator is strongly protected, with all later collaborative voices moving concentrically outward from it as comments, comments on comments, comments on comments on comments, and so on. The *Talmud*, a written record of multi-generational oral discussion, is the essential example of this collaborative model. Figure 41 compares examples of the encapsulated and collective models of collaborative authorship.

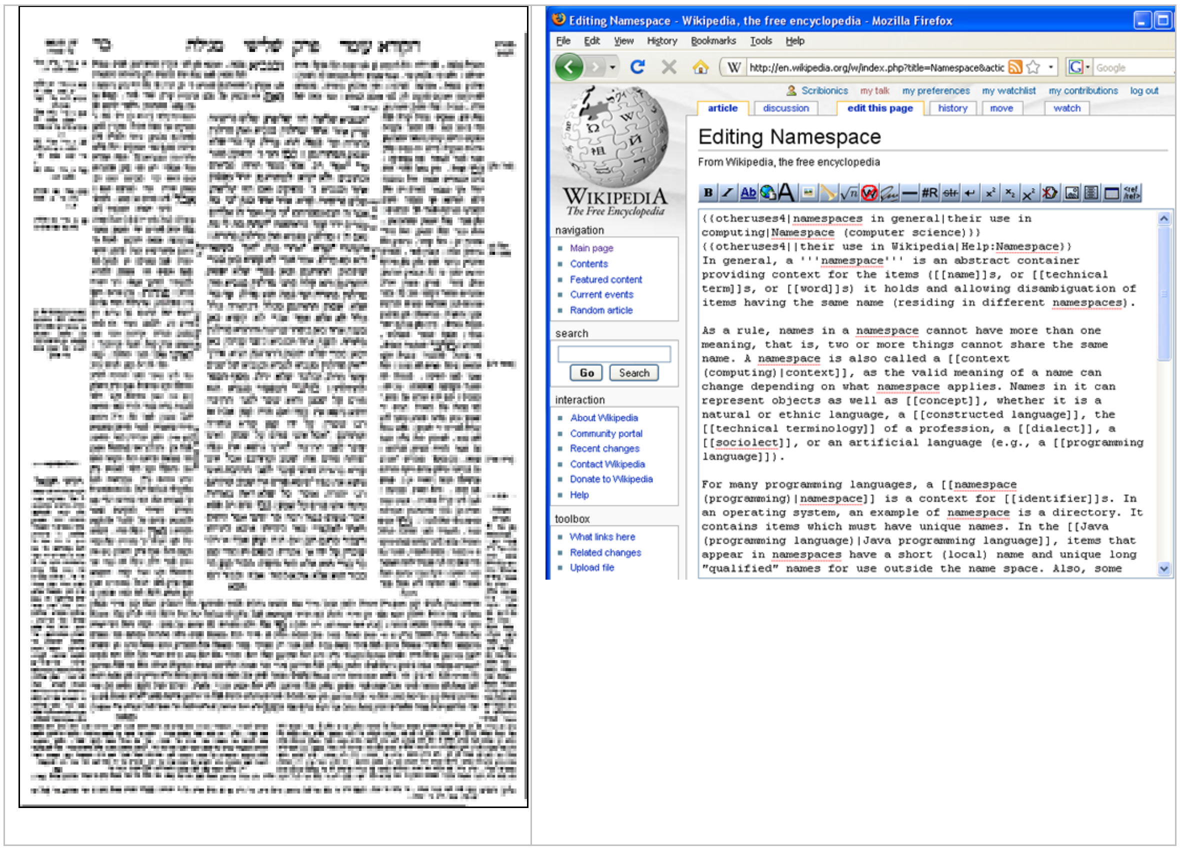
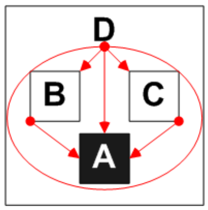
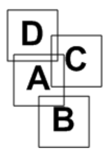
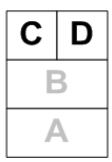


Figure 41: A Talmud page (L) protects the central text; a wiki page (R) exposes the central text to change.

In this example of a *Talmud* page (online at enlarged size at Eliezar Segal's www.ucalgary.ca/~elsegal/TalmudPage.html), the central text is surrounded by layers of commentary; the design of such a page allows commentary to fill the space surrounding the central text, overflowing to additional pages if necessary; ideas are fitted into page-shaped containers because this shape supports book manufacturing, not because the divisions between pages relate to divisions between ideas. As a digital text, the wiki page shown here is not constrained by the physical shape of paper: the "discussion" tab contains commentary separately from the central text; the "article" tab presents a reader-oriented view of the central text; the "edit this page" tab presents a writer-oriented view of the same central text, complete with tools for formatting and hyperlinking; the "history" tab contains an automatically-generated record of changes to the central text, complete with tools for undoing those changes. Later in this chapter, the "Directions of Movement" section examines bi-directional change in this page of the Wikipedia.

Table 10 introduces four additional models of collaborative textual construction.

Table 10: Four models of collaboration: focused, holistic, inspired, layered.

model	name	digital example	print example
	**focused**	chat hosted within larger community site	featured topic in scholarly journal
A+B+C+D	**holistic**		multiple authors of scientific journal article
	**inspired**	remix, sampling	parody, sequel
	**layered**	Content Management System (CMS) such as WordPress	book production

Special-topic issues of paper-like scholarly journals rely on the *focused* model of collaboration: an editor introduces the topic and the participants invited to discuss it; a featured author discusses the topic at length; several responding authors, provided with pre-publication access to the featured author's text, comment on the featured author's text as well as on the topic in general; an editor, provided with pre-publication access to all the texts, summarizes and responds to the full discussion. For example, in the October 2007 issue of *PMLA*, "Remapping Genre" is the special topic, introduced by an editor; Ed Folsom's essay on "Database as Genre: The Epic Transformation of Archives" is featured; five respondents comment on Folsom's remarks; Folsom comments on the respondents' remarks; an editor draws conclusions from the entire discussion. Although their work is presented together, within the larger container of the journal issue itself, the contributors do not work together but in parallel, each addressing the same topic independently; the author of every contributed component is publicly identified.

Collaborators operating under the *inspired* model may be more than independent: if an original work is expanded upon in ways the original author does not support, they may see each other as opponents, pulling against each other to shape the larger story in mutually-incompatible directions. For example, Margaret Mitchell's 1939 *Gone With the Wind* inspired both authorized (*Scarlett* (1991), *Rhett Butler's People* (2007)) and unauthorized (*The Wind Done Gone* (2001)) continuations by other writers. The later stories partially overlap with the original, re-using some of its characters, settings, and plot, but the majority of their content is unique.

The *holistic* model of collaboration is widely used in paper-like reports by journalists and scientists: multiple contributors may be named, but their individual contributions are not publicly tied to them in the finished text; within the community of collaborators, one person may be remembered as having added or improved a paragraph, but that memory is not available outside the group. Because hypertexts are assembled of separately-stored components, even if the individual contributor of each component is not identified at the surface level of the text, that information can probably be retrieved from sub-surface sources such as the database, file properties, and comments. By contrast, *layered* hypertextual collaborations do acknowledge separate contributions at the surface level, although those acknowledgements may recognize groups as well as individuals: Figure 42 shows a reader's view of a WordPress blog through a Web browser, in which three of the four named contributors (WordPress, MyMobiles, and Mozilla) are actually groups which themselves represent collaborations.

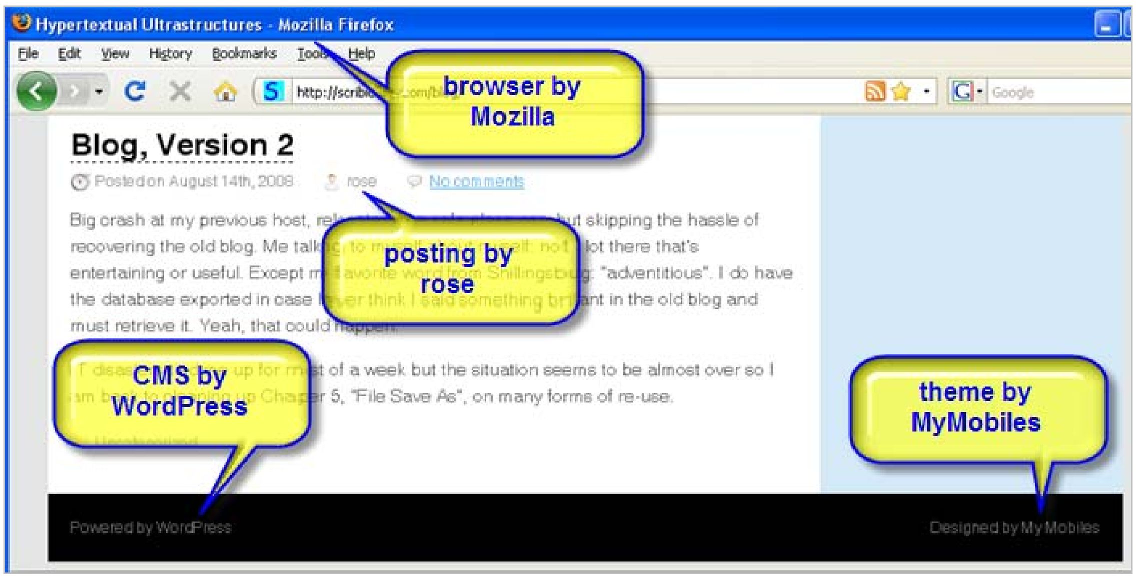
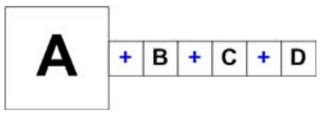
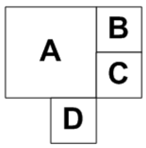
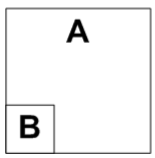
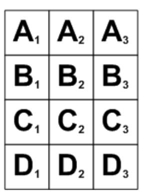
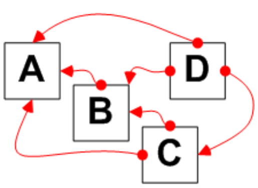
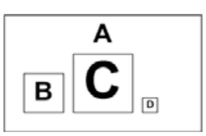


Figure 42: In a layered collaboration, contributors of broad categories are identified.

The groups contribute to the blog's construction in broad categories such as "design"; more granular information, relating the individuals within the group to distinct components within the broad category, may be available at sub-surface levels. For example, the README.txt distributed with this MyMobiles design identifies ten individuals who translated the theme's originally-English messages into their own languages: the translation into Italian ("ricerca nel sito" rather than "search the site") was contributed by Gianni Diurno, whose own blog is at www.gidibao.net

Table 11 introduces a third group of models of collaborative textual construction.

Table 11: Six models of collaboration: moderated, modular, personalized, structured, threaded, weighted.

model	name	digital example	print example
	**moderated**	blog + comments	essay + Letters to the Editor
	**modular**	software	encyclopedia
	**personalized**	dynamic Web page	form letter
	**structured**	database	telephone directory
	**threaded**	forum, game	chess game by mail
	**weighted**	social bookmarking	bookstore window

In a *moderated* collaboration, one or several of the participants have power to direct the discussion; in a hypertext, this can mean the power to alter or delete components of the discussion, creating instability within the collaborative text by continually revising it, often to meet the group's standards of word choice and content. Later in this chapter, the "Points of Control" section examines the different scopes of power assigned to anonymous guests, logged-in users, and administrators in a Simple Machines forum. The opposite approach toward maintaining group-wide standards is implemented in the *modular* collaboration of software development projects: rather than the ongoing efforts of a moderator, pushing non-conforming text back toward the group's established norms, modular software development works on the assumption that all contributors follow the guidelines initially established by a designer. Programmers can build components independently because they all abide by the same rules; modules that violate the rules are likely to fail when integrated into a larger whole.

In a *personalized* collaborative text, one voice or team of voices constructs a framework for the general case; another voice, operating within the framework, adds personal details and preferences which, combined with the framework, create a unique result. This model of collaboration is widely adopted in the online presences of shops, libraries, schools, and other enterprises in which the public institution (offering a catalog of choices) and the private individual (self-describing, and then selecting items from the catalog) cooperate to create a single-use, private text such as an invoice or a transcript. Implemented in dynamic Web pages, personalization is not a matter of navigational choice among pre-existing pages, which I maintain is not an act of authorship; in dynamic Web pages, expressed choices result in the construction of new pages, uniquely designed to a single visitor. Later in this chapter, a case study of the dynamic Web pages that create a shopping cart system examines this model of collaboration in depth.

While a personalized collaborative hypertext may be ephemeral, useful just for the moment of submitting a request, a *structured* collaboration exists for the purpose of long-term storage. In a structured database, records created independently are stored in consistently-shaped containers; each collaborator can provide unique information, but the kind of information accepted from each collaborator is the same. If some contributors can store free-form text, that is because the database structure includes a container designed to accept free-form text from all contributors. The "Access to Structure and History" section of this chapter examines some of the implications of this model of collaboration in a MySQL database.

Threaded discussion is the model applicable to some of the earliest-adopted and most widely-used hypertextual collaborations such as bulletin boards and discussion forums. This model is characterized by turn-taking and an assumption of shared history: the full discussion is available to all readers, even those who arrived long after the initial exchanges. A moderator can choose to lock an obsolete discussion, preventing latecomers from participating as writers, but nothing in the collaborative model requires that the growth of a discussion ever cease.

Social-bookmarking sites such as www.digg.com follow a *weighted* model of collaboration: as a component of the collaborative text becomes more popular, the pre-eminence of its position within the collaboration increases. In this model, most collaborators contribute nothing other than their vote of approval or disapproval: a vote to “Digg It” moves an item higher in the rankings, making it more visible and thus more likely to receive additional votes to “Digg It.” The purpose of the combined text is to announce community members’ reactions to component texts. Online newspaper publishers routinely offer readers a tool with which to register a “Digg It” for their favorite articles; if publishers complete the cycle by monitoring their publications’ ratings and seeking to produce more items resembling those that are highly rated, the larger project succeeds in shaping the content of the Web to suit its own collective preferences.

This list of ways in which texts, especially hypertexts, can be collaborative is not exhaustive: other models of collaborative textual construction may exist now; additional models will certainly exist in the future.

Hierarchies of Collaboration: Multiple Models, One Text

No one is in control of the Web: anyone who abides by the requirements of the networking architecture can publish a Web site. However, absence of central control does not mean absence of local control, nor absence of a hierarchy within which some sites exercise power and others seek to direct that power: some Web publications can help other Web publications gain visitors, creating a natural hierarchy in which some publications are gatekeepers on visitor’s paths to other publications. The high social and economic value of collaboration on the Web creates a strong concern with each Web publication’s place in the collaborative hierarchy; multiple simultaneously-valid perceptions of the collaborative natures of the same text are shaped by the observer’s position in the collaborative hierarchy.

For example, a reader of a *moderated* blog may be highly aware of its strong control by a single voice, with other voices sometimes allowed to provide minor, supporting commentary; awareness of the moderator's gatekeeping power is especially true if the reader wishes to participate as a writer. The blogger, though, may be most concerned with ensuring the blog's inclusion in higher-level collaborations: if the blog is included as a member of *aggregated* collections such as www.blogcosm.com and proves its popularity in *weighted* collections such as www.digg.com, more visitors will be attracted to the blog, bringing with them whatever economic or social value a visitor brings.

The webmaster who manages the site containing the same blog may think of it as conforming to the *layered* model of collaboration, a structured combination of software, design, and content. The site's database administrator (the webmaster for a small site, or possibly a separate specialist for a large enterprise), interested not in what ideas are explored in the blog nor in how they are presented to visitors, focuses on the efficient and reliable operation of the database in which blog postings are stored; from this standpoint, the entire project follows the structured model of collaboration, with each posting assembled of separately-stored components.

All these perspectives are valid, supporting my argument that a hypertext is must appropriately examined at multiple, simultaneous levels. What a hypertext *looks like* is only a thin skin at the top layer of what it *is*; *hyper*-collaboration is equally multi-layered.

Layers of Control

In Chapter II, I argued that digital reproduction, while relatively error-free for single texts, becomes increasingly error-prone when multiple independently-created texts are aggregated. Here, I argue that control over Web-based publication moves in the same direction: someone runs every Web site, but no one runs the Web. As collaborators are added, control is weakened until, at the level of the Web itself, a writer's experience is of complete freedom from the editorial policies, strict schedules, and centralized production that characterize print-based publication. This pattern of central control weakening as the number of participants increases applies not only to the Web but to many democratic projects; it can apply to print-based publications such as anthologies, in which authors of some items in the collection insist on exemption of their work from the editor's general standards.

For the one or several members who hold positions of power in a community constructing a digital hypertext, the added *hyper*-control attribute is in the ability to regain control after it has been released. For example, the moderator of a blog or forum can allow comments to be posted freely, without the moderator's approval or participation; later, if a comment is judged to be inappropriate, the moderator can remove it, leaving no surface-level evidence that it was ever published. If the comment is sufficiently unwelcome, its author's identity can be deleted, along with all the author's other contributions; the author can be temporarily or permanently banished from the community, blocking future misbehavior.

Example: Guests, Users, Administrators in a Simple Machines Forum

Threaded discussion in a forum operates on the basis of progressively-narrowing scopes of control. In Figure 43, “expectations for behavior in this forum” is a topic posted in the “policies” category of the “ADMINISTRATION” board of the “Journal of Topical Formulations” forum, which I moderate for a client; different readers of the same text have different kinds of power to use and change it. As the administrator, I have extraordinary power within the forum’s community of collaborative authors. For example, I can delete another author’s posting, or I can lock it so no other member of the community can publish a response to it. The forum is powered by open-source software available from Simple Machines at www.simplemachines.org; the right-hand side of Figure 43 compares the tools Simple Machines Forum makes available to an administrator, a non-administrative logged-in user, and an anonymous guest.

Figure 43: An administrator's view of a forum posting includes tools for deleting or altering it.

Some tightly-controlled forums allow only logged-in users to read their content; I have configured this forum so that anonymous guests may read but only authenticated (logged-in) members of the community may write. The scopes of control are narrowest for visitors who provide the least information about themselves: guests control their own movements and reading, and can use a tool to write (print) for their own private use; users can use tools to write (reply) and interact with other readers (send, poll, email, IM) and ask the system to interact with them (notify); administrators can use additional tools to insist that others read (sticky), prevent others from writing (lock), or undo (delete, remove) or alter (modify) or relocate (split, merge, move) others' writing. A guest can *observe* the discussion. A member of the forum can *participate* in the discussion. An administrator can *control* the discussion.

Directions of Movement

Forums, wikis, blogs, and other forms of digital hypertext are designed to coordinate multiple authorial voices into one collaboratively-created, evolving text. The component texts assembled into the larger collaborative text are stored in databases; databases are the topic of the next section on "Access to Structure and History." In this section, examination of a portion of the revision history for one wiki page will demonstrate several *hyper*-collaborative features available to communally-authored digital hypertexts of this kind. First, authorial identity operates differently for the text as a whole than for its separate components: lack of a named author of the top-level text does not imply anonymity, as members of the writing community are identified and permanently tied to their component contributions. Second, members of the writing community need not be human, and their contributions need not be constructive: the collective history of the collaboration tracks the work of natural and artificial intelligences as well as the work of vandals, who may be natural or artificial. Third, contributions of writing *per se* are a small fraction of the effort required to construct and maintain the text: besides adding content, contributors participate by repairing errors in content, repairing errors in structure, repairing vandalism, adding connections to the larger community, and improving adherence to community standards. Fourth, construction is chronologically bi-directional: new material necessarily moves the text forward in time; repairs can move the text forward or backward in time.

Example: Wikipedia Page Constructed by Humans and Robots

As pointed out above in introducing the collective model of collaboration, every wiki page customarily maintains an automated history of its own construction, available on the page's "history" tab; Figure 44 is a portion of the history page associated with the Wikipedia page explaining the concept of "namespace," at en.wikipedia.org/wiki/Namespace.

11:47, 3 June 2008 Chobot (talk | contribs) **m** (2,275 bytes) *(robot Adding: ko:이름공간)* (undo)
04:28, 1 June 2008 DHN-bot (talk | contribs) **m** (2,255 bytes) *(robot Adding: vi:Không gian tên)* (undo)
18:19, 28 May 2008 Robbot (talk | contribs) **m** (2,231 bytes) *(robot Adding: es:Namespace, zh:命名空间)* (undo)
02:53, 2 May 2008 Mdd4696 (talk | contribs) (2,194 bytes) *(rm Wikipedia self reference)* (undo)
05:19, 25 March 2008 Stusutcliffe (talk | contribs) (2,273 bytes) *(stub sorting)* (undo)
14:54, 20 March 2008 Thumperward (talk | contribs) (2,264 bytes) *(stubify)* (undo)
18:50, 18 February 2008 Ossmann (talk | contribs) **m** (2,253 bytes) *(Reverted edits by 207.106.1.34 (talk) to last version by ClueBot)* (undo)
18:49, 18 February 2008 207.106.1.34 (talk) (empty) *(←Blanked the page)* (undo)
18:49, 18 February 2008 ClueBot (talk | contribs) **m** (2,253 bytes) *(Reverting possible vandalism by 207.106.1.34 to version by Scribionics. False positive? lueBot. (226138) (Bot))* (undo)
18:48, 18 February 2008 207.106.1.34 (talk) (empty) *(←Blanked the page)* (undo)
00:54, 25 January 2008 Scribionics (talk | contribs) **m** (2,253 bytes) *(consistent verb form: providing context,…allowing disambiguation)* (undo)
03:02, 22 January 2008 Algebra (talk | contribs) (2,251 bytes) *(rm extraneous punctuation)* (undo)

Figure 44: Six-month history of the English-language Wikipedia page on "namespace."

In this record of changes to the wiki page between January and June of 2008, collaborators include humans and robots; in Wikipedia, a "robot" is software, without a physical structure. Sometimes a robot is specifically directed by a human to perform a tedious task of which the human is aware, such as searching out all the uses of a term wiki-wide and changing them to use an alternative term. The Robbot, DHN-bot, and Chobot robots changing the page May 28, June 1, and June 3 may have been operating in this human-supervised way, adding hyperlinks between this English-language page and its parallel pages in Spanish, Chinese, Vietnamese, and Korean (es, zh, vi, ko). A robot can also act without specific human direction, in response to events it is monitoring within the text. For example, at 18:48 on February 18 the content of the "namespace" page was blanked out, leaving an empty container; one minute later, the ClueBot automatically returned the page to its previous state; in the same minute, the same vandal emptied the page again; one minute later, a human "Recent Changes

Patroller," notified of the struggle between the ClueBot and the unnamed vandal at IP address 207.106.1.34, manually returned the page to its previous state; the struggle then ended, with no further changes to "namespace" until March 20.

Robot and human participants in the wiki's writing collaboration can describe themselves or be described in user pages such as en.wikipedia.org/wiki/User:ClueBot and en.wikipedia.org/wiki/User:Ossmann, describing two of the participants in the "revert war" discussed above. Human participants can focus on specific interests and self-assigned roles within the wiki, performing narrowly-defined functions just as the robots do; in their user pages, some of the humans listed above describe themselves as "Recent Changes Patrollers" or "Stubifiers" or aspiring "Administrators." While no user page in the Wikipedia describes the vandal at 207.106.1.34, the source of the disruption can be identified: publicly-available WHOIS records locate IP address 207.106.1.34 in Fleetwood, Pennsylvania, in a range of addresses managed by PaeTec Communications, an Internet Service Provider (ISP) with corporate offices in Fairport, New York and at www.paetec.com. Abuse of PaeTec's services can be reported by toll-free telephone or at abuse@paetec.com. WHOIS data, which must contain instructions for reporting abuse, is easily obtained from sources such as IP-Lookup at www.ip-lookup.net; like the StatCounter data discussed under "What the Webmaster Saw" in Chapter IV, this supports my argument that anonymity on the Web is only a surface-level phenomenon.

The "namespace" page under discussion here was created October 18, 2001 and has since been changed by hundreds of collaborators; focusing on the six months of its history sampled in Figure 44, Figure 45 visualizes the page's evolution.

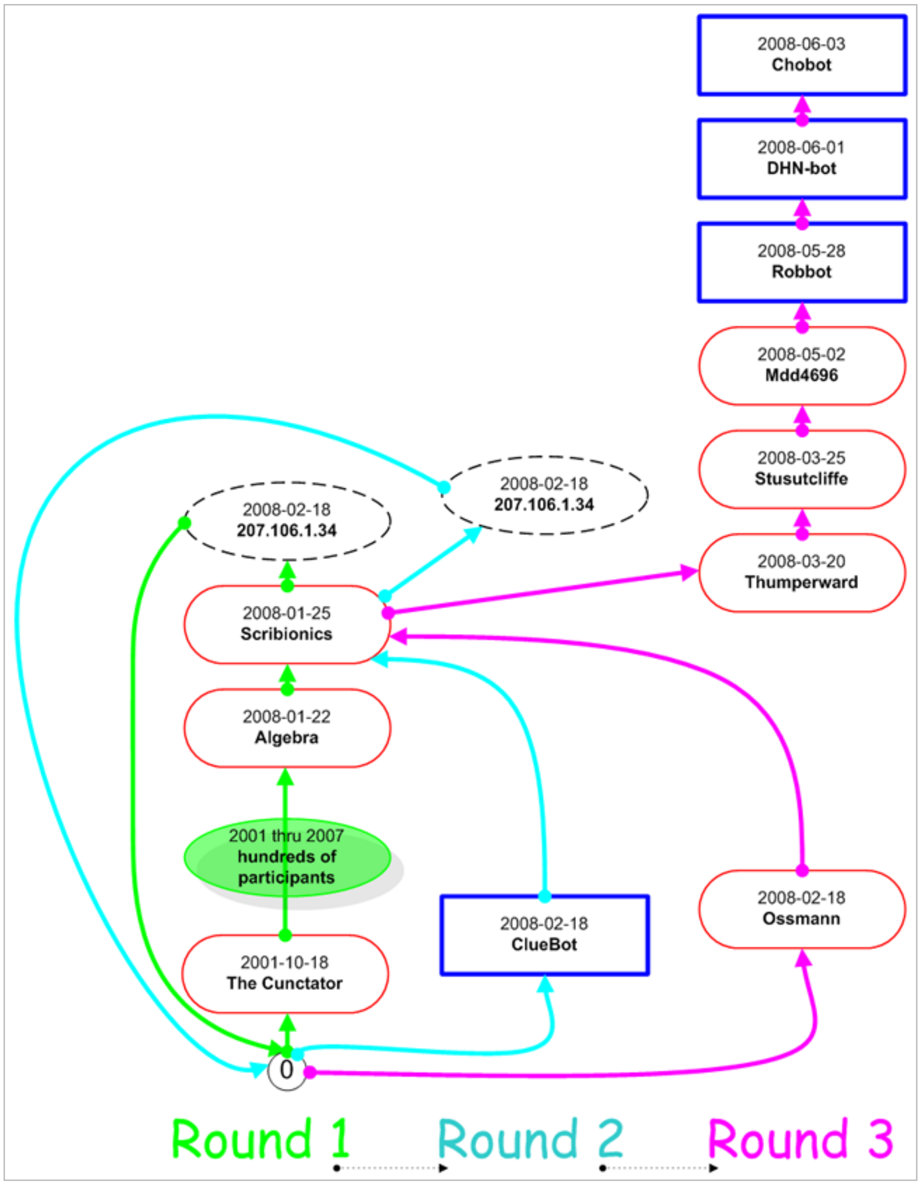


Figure 45: Collaboration in a wiki page can move the text forward and backward in time.

Here, robotic collaborators are represented by blue-outlined rectangles; self-identified human collaborators are represented by red-outlined rounded lozenges; the unidentified vandal is represented by a broken-lined ellipse. In Round 1 of the page's evolution, it moved forward in time from its creation in 2001 until it was vandalized in 2008, at which point it moved backward in time to before the point at which it was created. Round 2 lasted less than one minute, moving the page forward to the point before it was vandalized and then immediately back to before it was created. Round 3 returned the page to forward movement in time, again moving it ahead to the point before it was vandalized, from which normal development by humans and robots resumed.

Bi-directional movement of this kind is possible because all versions of the wiki page are permanently stored in a database. Beyond simple recordkeeping, permanent access to previous versions enables wiki editors to publish a change, observe reader reaction to it, and then un-publish the change; while a word-processing tool's Edit—Undo tool allows an author to privately examine and reconsider change only while the editing session is in progress, a wiki's Revert function enables an author to benefit from and respond to public examination and feedback, with no limitation imposed by distance in time from the moment of the change.

Persistent Access to Components

If knowledge is power, then the keeper of a database in which knowledge is stored has some *hyper-* version of power: not only *access* to what is currently known, but *control* over what can be known in the future.

Example: WordPress Blog Postings Stored in MySQL Database

Figure 46 shows a portion of the MySQL database in which the same WordPress blog discussed in Chapter III is stored. The content of the WordPress blog which can be visited at www.scribionics.com/blog is stored in a database named wrdp1. The database consists of ten tables, separately containing blog components such as postings, comments, and user accounts. One of the tables, wp_posts, contains the blog postings themselves, storing each one as a record in the table; the grey-shaded columns of Figure 46 shows three such records, containing three revisions of one posting (p=36, p=37, p=38), titled “the publisher has no control.”

ID	36	37	38
post_author	1	1	1
post_date	2008-12-15 18:35:22	2008-12-15 18:34:45	2008-12-15 18:35:22
post_date_gmt	2008-12-15 23:35:22	2008-12-15 23:34:45	2008-12-15 23:35:22
post_content	Finally cut out a good chunk that I think is a dif...	Finally cut out a good chunk that I think is a dif...	Finally cut out a good chunk that I think is a dif...
post_title	the publisher has no control	the publisher has no control	the publisher has no control
post_category	0	0	0
post_excerpt			
post_status	publish	inherit	inherit
comment_status	open	open	open
ping_status	closed	closed	closed
post_password			
post_name	the-publisher-has-no-control	36-revision	36-revision-2
to_ping			
pinged			
post_modified	2008-12-15 18:35:54	2008-12-15 18:34:45	2008-12-15 18:35:22
post_modified_gmt	2008-12-15 23:35:54	2008-12-15 23:34:45	2008-12-15 23:35:22
post_content_filtered			
post_parent	0	36	36
guid	http://scribionics.com/blog/?p=36	http://scribionics.com/blog/?p=37	http://scribionics.com/blog/?p=38
menu_order	0	0	0
post_type	post	revision	revision
post_mime_type			
comment_count	0	0	0

Figure 46: Database records contain three revisions of a blog posting.

Each record is organized into twenty-four fields; the post-title field contains the title, identical in all three records; the post_content field contains the words of the posting itself, slightly different in each record. The post-name field places the three records in their proper relationship to each other: 36 is the original, internally named to resemble its surface-level title (the-publisher-has-no-control); 37 is named to relate it to 36 (36-revision), as is 38 (36-revision-2). All the fields of all the tables in the database can be edited by the database administrator, either by writing unique instructions to that effect in the MySQL query language or by using the visual interface of the phpMyAdmin database administration tool; Figure 46 shows the partial results of phpMyAdmin's Structure—Browse function. By manipulating the database directly, the database administrator can alter or delete the stored revisions, including descriptive components such as timestamps and the author's userid.

The Narrator's Voice in Dynamic Web Pages: Personalization, Localization, Visible Seams

In any text, gaps in language are symptomatic of gaps in thought: a mismatch in the text suggests a disconnection in its construction process; someone changed the plan, or a new plan, perhaps the work of a new person, replaced the original. Visible seams within a text, indicating points at which originally-separate components have been imperfectly assembled, can support hypotheses that the text has survived changes in its structure: an author's note that a never-before-mentioned idea was introduced "above" hints that something intended for the text was forgotten or deleted or relocated from its original place, as does an unfulfilled promise of elaboration "below."

Unlike the author of a printed text, the author of a hypertext cannot validly assume that all of the text's readers will have similar experiences of "above" and "below," but structural disconnections can still be observed. Close reading of the surface level of a hypertext can identify evidence of assembly and collaboration, suggesting reasonable hypotheses about how the text was constructed. An instruction to "click OK to proceed" on a page that offers nothing labeled "OK" but does offer a button labeled "Continue" hints at disconnection between the creators of human-language messages and language-labeled buttons. Similarly, following a hyperlink from a page of black text on a white background to a page of blue text on a yellow background suggests that, whether or not these pages were actually created by the same author, they are to be read as if they were disconnected, artificially joined by the hyperlink but not meant as continuations of the same thought; whatever commonality prompted the hyperlink is too weak to support a shared design.

Below the surface, close reading of a hypertext's ultrastructure can explore not only *where* divisions exist within a hypertext, but *why*. As shown in the following case study, the common programming practice of addressing human communication—messages—separately from a computer's activities—logic—allows a system to offer each user a customized experience within a rigid and internally-consistent structure. One way in which dynamic Web pages accomplish this is by allowing each human visitor to select a preferred form of communication: visitors might be asked to select from a list of available human languages or indicate their own level of expertise—novice or advanced—so that the system might formulate messages appropriately. In this way, the "same" hypertext can speak in the crisp voice of an expert colleague, or the cheerful voice of a patient teacher, or the colloquial voice of a native speaker of the visitor's own language—assuming, of course, that the voice of such an expert or teacher or native speaker has been captured and the system has been directed to use it. The case study that follows, taken from my own practice, illustrates the method by which dynamic Web pages—in this case, as part of an online shopping system—are enabled to choose a human language and speak in it.

Case Study: Human Language as User Preference in Shopping Cart Software

On the Web, the shopping cart is a ubiquitous metaphor for expression of an individual's preferences within an institution's catalog of possibilities: the individual places items in the shopping cart to indicate initial interest; later, the items can be discarded from the shopping cart if the interest fades, or permanently acquired (possibly purchased via credit card) if the interest endures. In this way, the shopping cart itself becomes a container of the shopper's selections and personal identity (credit card number, shipping address), within the larger context of the institution's offerings and policies. A variety of software tools are available which can provide Web site visitors with the mechanisms of selecting items and purchasing them with a shopping cart; in my own practice as a webmaster, I use the communally-created osCommerce, available as open source from www.oscommerce.com.

Because osCommerce is an open-source tool, as are its primary components—the PHP programming language and the MySQL database language—its internal structure is readily accessible for thorough examination and, as necessary, modification. The structure and behavior of osCommerce make it useful here as an example of Web-based software that containerizes communication with humans separately from other activities of the system, preserving and exploiting the separation between meaning and presentation. The screenshots and code fragments I examine in this case study reflect my own in-progress customizations to a single site, www.new-equilibrium-skincare.com, a seller of skin and hair treatments.

For purposes of demonstration here, I will discuss communication related to the widespread and very elementary process of separation from a Web site, also known as logout, logoff, signout, signoff, exit, or de-authentication; in this process, a visitor who has previously entered a section of the site which requires privacy and an authenticated identity (for example, to enter a credit card number) expresses the desire to end that private session and return to a publicly-available view of the site. Figure 47 shows the sample site's logoff page; close examination of the ultrastructure of this page demonstrates some of the methods by which highly-containerized systems provide a user experience that is variable and dynamic, allowing the visitor's expressed preferences to shape much of the content of the Web page. This also demonstrates how the institution's voice can be changed: in this example, the English-language messages as supplied with the software were written in passive voice ("Your order has been received") but the client prefers to speak in active voice ("We received your order"); because osCommerce stores human-directed messages separately from computer-directed logic, making this change system-wide required changes to message files but not to programs.

Figure 47: User-selected human language (English) placed in software-defined containers.

The general structure of the logoff.php page is consistent with all the pages in this site. The page is subdivided into five sections: a header, a footer, columns on the left and right, and a large central area in which the main message of the page is communicated. To maintain site-wide consistency, the content of each section is managed separately; for example, every page presents the same *footer* by calling the same program, footer.php, which in this case provides some variable status information (the current date), some fixed legal information (a copyright claim), and fixed hyperlinks to sites related to the shopping software (osCommerce and Google Checkout) and to the manufacturer of the products sold on the site (Elsom Research).

Similarly, the *left column* is the same on every page, offering every shopper the same hyperlinks to pages that offer products or explain policies, as well as a box for entry of keyword search terms. The shape of the *right column* is the same on every page and for every shopper, but its content varies in response to activity on the site: the ranking of bestselling items changes as orders are processed. The content of the right column also varies in response to activity of each shopper: this shopper's shopping cart currently contains no items; another shopper at the same moment might have selected several items, in which case the "Shopping Cart" container in the right column will list those items on any page that shopper displays. The *header* also contains static and dynamic information which appears on every page in the site. The manufacturer's logo is hyperlinked to the manufacturer's home page; this image, in the same location and with the same hyperlink, is present on all pages of the site. Elements on the left side of the header visibly adjust to shopper activity: the hyperlinks to Top >> Catalog >> Log Off, called a "breadcrumb trail," identify this page and the main path to it, varying appropriately as other pages within the site are visited. Elements on the right side of the header present a consistent appearance to all shoppers: everyone always sees hyperlinks to "My Account," "Cart Contents," and "Checkout," and everyone who is logged on also sees a hyperlink to "Log Off." The appearance of these hyperlinks is unchanging, but they behave in a manner that is unique to each shopper: a hyperlink to "My Account," for example, is presented to every shopper in the header of every page, but only that shopper's personal account information is displayed if that hyperlink is clicked.

The center or *main* section of the page seems as if it should never vary, presenting the same information to every user who causes the logoff.php program to execute. Logoff.php does perform only one function, constructing a page in which the main section uses human language to confirm that the user's request has been processed ("We logged you off") and advise the user how to proceed ("You can now safely leave"). However, examination of the ultrastructure defined by logoff.php shows that even this simple, predictable act of communication is meant to be customizable for each user; this is possible because the osCommerce user interface is predicated on a strict separation between container and contents.

Figure 48 relates user selections of action (logoff) and language (English) to the system's placement of the English-language contents of TEXT_MAIN in the main section of the logoff.php Web page.

Things to Do	Ways to Speak While Doing	Places to Speak While Doing	Things to Use While Doing	Things to Say While Doing
-	-	header	-	**TEXT_MAIN**
-	**English**	footer	\includes\languages\	HEADING_TITLE
-	-	left column	-	NAVBAR_TITLE
-	-	right column	-	
-		**main**	-	
-		heading		
-		navbar		
-				
logoff \logoff.php				

\includes\languages\english\logoff.php

```
define('HEADING_TITLE', 'Log Off');
define('NAVBAR_TITLE', 'Log Off');
define('TEXT_MAIN', 'We logged you off from your account.
You can now safely leave your computer.<br><br>
We saved your shopping cart;
the items you selected will be remembered when you log back into your account.');
```

Figure 48: A visitor's choice of action (logoff) and language (English) dictates the content of the Web page.

Here, the logoff.php program defines a Web page to serve as a container in which messages such as HEADING_TITLE, NAVBAR_TITLE, and TEXT_MAIN can be placed. The function of logoff.php is to create an environment which supports presentation of those messages; the messages themselves are defined elsewhere, in a location that is only referenced after the shopper’s preference of a human language is expressed. In this example, because the shopper has expressed a preference for communication in English and the webmaster has specified a separate directory as the source of English-language messages, logoff.php obtains the text of its messages from \includes\languages\english\logoff.php. The osCommerce software is structured to be infinitely extensible in this way, supporting communication in any human language; by setting up the required messages in other languages, and allowing each shopper to choose from among those languages, the text presented by logoff.php can vary in surface appearance (speaking English or Spanish or any configured language) while its essential meaning (assuming equal competence among the human authors of the messages in each language) remains constant and its structure (as defined by logoff.php), which exists once but can present messages in any language, is unaffected by linguistic choices.

Systems like osCommerce, structured to allow multiple equally-valid voices to speak, demonstrate Russian formalist critic Mikhail M. Bakhtin's idea of *polyphony* as a narrative technique. In a polyphonic narrative, voices do not express competing ideas, not all of which can be true; nor do they express complementary ideas, each containing some part of the truth and capable of assembly into a larger whole which communicates the full truth; instead, every voice expresses the complete truth, but expresses it in a unique way appropriate to the speaker's position in the narrative (Haynes 296). Bakhtin, who died in 1975, applied his ideas about polyphony to works of printed fiction; I argue that, even in non-fiction digital hypertexts such as the online shopping site in this example, there is a central, guiding voice. This voice communicates in human-language messages which, like the narrator's voice in a work of fiction, are crafted by a human author to shape the human reader's experience of the text. When, as in this case, the reader can select a narrator from among several available voices (and alter that selection at any point in the reading if the original choice is unsatisfactory), a hypertext has a *hyper*-narrator. As discussed previously, I argue that hypertexts are like poetry in that their structure is a dominating influence, and like drama in their inherent separation between on-stage performance and back-stage control; I further argue that hypertexts are like fiction in their central positioning of a narrator’s guiding voice, and that fiction-related studies such as narratology may be naturally expanded to address questions related to communication between the voice of a hypertext (offering advice and interpreting complexities) and a reader’s understanding of that hypertext.

Every point at which osCommerce communicates with human visitors is structured in the same way to support the visitor’s choice of a guide’s voice. Each program points to the container of its messages, within the container of all messages in the visitor-selected language (\english\, in this case), within the container of all languages (\languages\), within the container of all external components to be used by osCommerce programs (\includes\): English-language messages within the osCommerce site are stored in \includes\languages\english\. Written in the PHP language rather than directly as HTML, logoff.php assembles the selected message into the space reserved for it, then uses the resulting HTML to format the Web page; use of external components in this way is the hallmark of dynamic Web pages. In one data cell of one row in one table, logoff.php calls for the content of the main section and identifies the formatting to be applied to it: <td class=“main”><?php echo TEXT_MAIN; ?></td>. The English-language content of TEXT_MAIN for logoff.php, as shown in Figure 48 above, provides the words to be presented on the page: We logged you off [...]. With the completed page displayed, if the visitor asks to see its internal makeup (View—Page Source), only the final combination is available; Figure 49 traces the steps to this result.

Figure 49: Multiple layers of substitution into placeholders allow human language to be presented with other user-specific data.

Intermediate stages of a dynamic Web page's construction—logoff.php calling for the contents of TEXT_MAIN—are not accessible through a browser's surface-level presentation of a finished Web page, nor even at the level immediately below the surface as made available by the browser's View—Page Source tool: by the time the text is available to the browser, all substitutions have taken place and the page is fully constructed, with "We logged you off" occupying the space held for it by TEXT_MAIN. A dynamic Web page, even more so than some other hypertexts, cannot be fully studied without access to all the layers and structures that contribute, even in a transitory way, to its final form. Surface-level observers, relying on a Web browser to provide access to a hypertext, do not have the necessary access; the webmaster, able to examine components of the site independent of the mediating effects of a browser, is in the ideal position to observe, comment on, and experiment with hypertextual construction.

Various Kinds of Variability: How Dynamic Is This Page?

Many Web pages are dynamic, presenting each visitor with a customized reading experience which may be irreproducible for a critic who wishes to share and comment upon that experience. As shown in the preceding case study of one very simple dynamic page, a page can be dynamic in a variety of ways; to contribute some meaningful understanding of a hypertext, a critic must comment not on the *fact* of a text's variability but on its *nature*, at a level of granularity consistent with that of the text's ultrastructure. In Table 12, I list the types of variability I observe in elements of the very simple dynamic page discussed above, logoff.php, by providing each type with a concise identifier and an example, and describing the scale on which it varies and the source (in some person at some time) of its variability.

Table 12: Granular description of multiple kinds of variability demonstrated in one dynamic Web page.

identifier	example	scope of consistency	source of change	time of change
CONSTANT	manufacturer's hyperlinked logo (in header)	site-wide	webmaster	design of header for all pages in site
ENVIRONMENTAL	timestamp (in footer)	site-wide for this moment	server's clock	page load
CALCULATED	number of items assigned to "Starter Sizes" category (in left column)	site-wide	webmaster	structure of product catalog
NAVIGATIONAL-TARGET	address behind "Starter Sizes" hyperlink (in left column)	site-wide	webmaster	structure of product catalog
NAVIGATIONAL-PAGE	breadcrumb trail (in header)	this page as displayed for any shopper	page load	page load
EVOLUTIONARY	ranking of popular products (in right column)	site-wide for this moment	aggregated actions of all visitors at all times	any shopper's purchase of item(s)
PERSONAL-POSSESSION	shopping cart contents (in right column)	site-wide for this shopper	this shopper's action during this session	this shopper's selection of item(s)
PERSONAL-MESSAGE	"We logged you off …" (in body)	this page as displayed for any shopper who prefers messages in English	this shopper's selection of language preference	this shopper's login
PERSONAL-TARGET	address behind "My Account" hyperlink (in header)	site-wide for this shopper	this shopper's login	this shopper's login
PERSONAL-BROWSER	fonts used in non-image text, if site defaults overridden	Web-wide for this shopper	this shopper's browser options	page load
EXTERNAL	Google Checkout's hyperlinked external logo	Web-wide, for all Google Checkout merchants who chose this logo	webmaster's selection or Google's alteration of this logo	page load

A more complex hypertext might demonstrate a wider variety of variability; developing a complete vocabulary for identifying and then criticizing all the ways in which hypertexts can be dynamic is a task that cannot be completed while technological advancements continue to expand the options available to creators of hypertexts.

Uncontrollable Collaborators: Reader-Induced Variability in "Constant" Components

Recognizing that all elements on a Web page, including items of type "CONSTANT," have some potential for variability is essential to understanding the nature of Web-based publication and how it differs from paper-based publication: the "same" Web page, even one that appears to consist solely of static elements, can have a different appearance for different readers. Everyone who borrows the same paper book from a library will see the same final page of the book; for logoff.php, the final page of its Web site, for the same shopper who makes the same choices the page will look substantially the same (excluding elements such as the timestamp and the product popularity rankings, which change for reasons no visitor can control) only until someone who controls the page—the webmaster or the designer—alters some component of its construction, at which point the next instantiation of the page will be changed (when constructed per request) for all other visitors.

The webmaster has central, but not exclusive, control over the appearance of the page. Visitors to a Web page, static or dynamic, can control many aspects of its appearance and behavior through options offered by their personal Web browser. The 2009 edition of the *MLA Handbook for Writers of Research Papers*, noting that electronic texts can be "accessed through a variety of interfaces displayed on different kinds of equipment," suggests that "accessing a source on the Web is akin to commissioning a performance" (181); multiple performances of the same play or concert are likely to strongly resemble each other, but unlikely to be identical.

Web browser software is designed to add a potential for variability to many otherwise-static page elements, especially in terms of colors and sizes; for example, Figure 50 illustrates the choices offered (Tools—Options—Content) by the Mozilla Firefox browser; Firefox users can decide whether to accept the Web page designer's preferences or to insist upon their own.

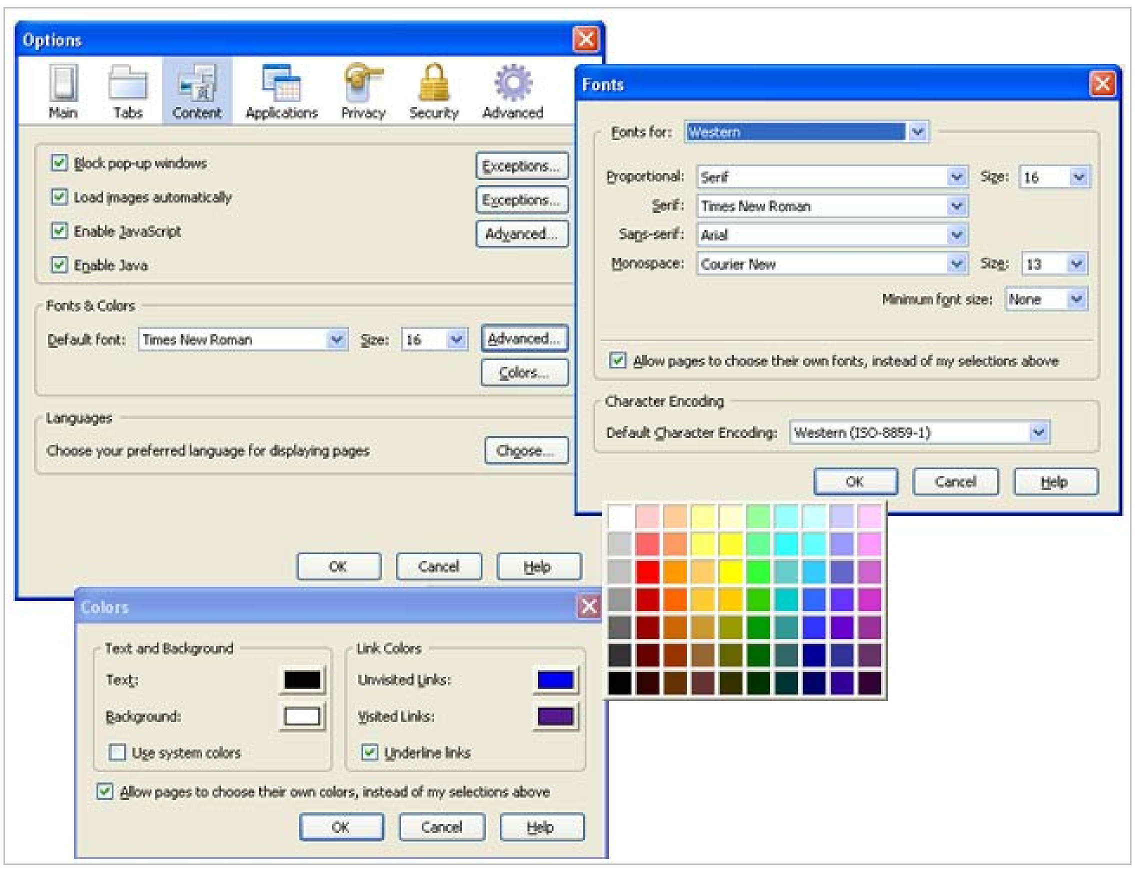


Figure 50: Firefox users optionally allow Web pages to use the colors and fonts specified by their designers.

For all the sites they visit, Firefox users can control elements of appearance— the size of text, the color of unvisited hyperlinks—and behavior—block pop-ups, load images, run scripts—specifying their own preferences to override those of the designer: “Allow pages to choose their own fonts, instead of my selections above” is an option, but far from the only option. Customization methods such as this expand the usability of Web pages, making a highly-visual medium accessible to visitors whose limiting visual requirements might otherwise prevent them from comfortably using it; they also mean that to a great extent the creator of a page controls only the default design of the page, the design presented to a reader who has not bothered to describe a preferred design—no images, larger fonts, fewer colors—which may be quite unlike the page as experienced by any other reader.

Uncontrollable Collaborators: External Components “Within” a Dynamic Web Page

In addition to variability caused by choices made by a designer or a specific reader, as well as changes caused by aggregated activities of other visitors or by factors as simple as the passage of time, many Web pages are exposed to variability of the type I have identified as “EXTERNAL” in Table 12. In the logoff.php example, the footer contains an image of the credit cards shoppers can use via the Google Checkout service to purchase items displayed in the site’s catalog. Unlike the other images displayed on the page, the image supplied by Google Checkout does not reside with the other components of logoff.php. Examination of the relevant portion of logoff.php’s ultrastructure, included in Figure 51, identifies the source of the image.

```
<!-- Start of Google Checkout Badge Code -->
<center>
<link rel="stylesheet" href="https://checkout.google.com/seller/accept/s.css" type="tex
media="screen" />
<script type="text/javascript" src="https://checkout.google.com/seller/accept/j.js"></s
<script type="text/javascript">showMark(2);</script>
<noscript><img src="https://checkout.google.com/seller/accept/images/ht.gif" width="182
height="44" title="Google Checkout Acceptance Mark" alt="Google Checkout Acceptance Mar
/></noscript>
</center>
<!-- End of Google Checkout Badge Code -->
```

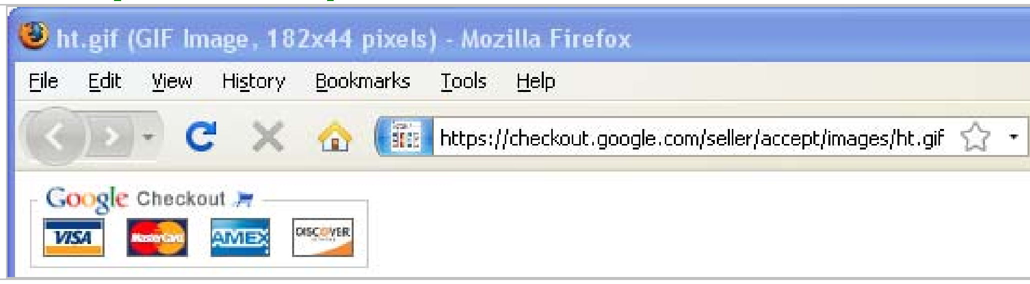


Figure 51: Code in logoff.php calls for externally-controlled components.

As shown here, the Google Checkout image, and the stylesheet that controls its appearance, and the JavaScript program that will execute if the image is clicked, all reside externally, at https://checkout.google.com/seller/accept. If the server that supports checkout.google.com becomes unavailable, either because of its own internal failure or due to some connectivity failure between it and the Web page visitor, those components of logoff.php will be unavailable, represented only by an empty container 182 pixels wide and 44 pixels high. More relevantly to any discussion of variability and control, external components such as these can be changed or eliminated by their own caretakers at any time, with no requirement for the webmaster or the visitor to take action to implement or allow the change, nor even to be informed of the fact of a change; these components appear on the surface of the page as if they are integrated into the page but they remain ultrastructurally separate from it, completely controlled by Google Checkout.

Toward Meaningful Descriptions of Hypertextual Collaborations

Compiling concrete data about Web page elements such as those identified above can produce a lengthy and detailed description of the page, similar to the descriptive bibliography that, for printed books, records details of the book's structure such as the length of its index and the number of illustrations it contains; like a book's descriptive bibliography, a detailed and granular description of a hypertext would be useful in comparing it to seemingly-similar hypertexts, seeking points at which they vary. Such a description could be read in the customary way, as words, but it could also be visualized using techniques such as the Kiviat diagrams (Figure 23) used to expose intentional changes in the Versioning discussion in Chapter III. Expressed in this way, the structures of large numbers of Web pages could be examined and compared in any terms considered useful for a given analysis. For example, identifying pages with a high reliance on components controlled by external collaborators (such as the Google Checkout badge in the preceding case study) is a good way of predicting which normally-online pages are least likely to operate correctly in offline situations, such as from a locally-saved private copy; similarly, pages which include personalized elements (such as the current user's name and contact information) must be most thoroughly checked for violations of privacy, possibly requiring some portions to be blocked or blurred before copies (such as screen shots used in training material) can be distributed.

Thorough analysis of a Web page's structure supports responsible creation and use of copies of the web page.

An appropriately granular, ultrastructurally-based descriptive and functional criticism of hypertext would begin by dividing a page into its components, considering what is "text" separately from what is "*hyper*-." At the surface level, there are at least containers and contents, each with their own design goals and their own innovations and inadequacies. At deeper levels, there may be containers within containers, and there are likely to be components (such as the "My Account" link from logoff.php above) with content that varies even when their appearance does not; a full description of a hypertext cannot be created without examining its internal makeup and its behavior. With full descriptions in hand, comparative studies become possible. If comparative studies of small numbers of hypertexts can be performed, using appropriate metrics in repeatable ways (for example, assigning numeric values to indicate the manner and extent to which each component of a page is dynamic), similar studies can be automated to address the large numbers of hypertexts that populate the Web.

Can a Hypertext Be a Solo Project?

Like software development, book production is a normatively-collaborative project although, unlike software development, most of those involved in producing a book are customarily unacknowledged. In addition to the ideas contributed by an author, a book is constructed of three things: a method of making marks; a surface on which to make marks; a method of fastening the marked surfaces to each other so they are recognized as one entity. The people who invent and produce these book-making components are not ordinarily named, collectively or individually, as contributors to the creative process; neither are the people who, with the author's ideas adding the final component, assemble book-making components into perhaps thousands or millions of printed copies of the finished text. Perhaps all these workers are anonymized because their contributions are seen as trivial, merely improving upon ancient and highly-accessible techniques; after all, even small children can construct books of their own drawings and stories. The world is filled with mineral-, animal-, and plant-based material adaptable to the purposes of making marks, being marked upon, and fastening together what has been marked upon; for an author determined to produce a book as a solo project, even to the point of creating the book's paper and ink and binding, seemingly-infinite options are available and await only a creative mind with time and ingenuity to combine raw material in unique ways.

Could an author also create a one-of-a-kind hypertext by combining available raw material? After all, digital documents are said to be easier to produce than paper documents; thanks to such features as copy-and-paste and multiple levels of undo, getting everything just right is less physically demanding and less time-consuming than writing and re-writing multiple manuscript drafts. The difficulties inherent in such a project are hinted at by two viewpoints expressed by novelist and teacher Robert Coover, an early adopter and critic of hypertext as a creative writing tool. In his 1998 *New York Times* essay "The End of Books," Coover enthusiastically praised hypertext as a method of expanding the creative options available to his undergraduate students, encouraging them to improvise and innovate beyond the bounds of paper-based writing;

a few years later, cited in N. Katherine Hayles 2002 *Writing Machines*, Coover expressed fear of the author's voice being "overwhelmed" by technology, and announced that the "Golden Age of hypertext was over and we were rapidly declining into the Silver Age, if not the Bronze and Iron" (44), in which no new ideas or accomplishments were possible, merely poor imitations of earlier innovation:

> Coover also expressed concern about the relentless cycles of software innovation and obsolescence. He felt he could not continue to master all the new software programs coming out at an accelerating pace and still devote his energy to what he cared about most, crafting words. (45)

Coover and others who initially embraced hypertext as an opportunity for expanded creativity are correct in observing that the demands of working in hypertext can complicate writers' access to free-form innovation, if that innovation depends upon solo control of all aspects of "book" production.

Coover is right about the pressures created by the "relentless cycles of software innovation and obsolescence." Hypertexts are mediated by software; software is designed not only to enable, but often to require, collaboration. The shelf life of any specific computer-related knowledge or skill is generally short; it is possible to learn enough to accomplish the task of a given moment, but the task of the next moment is likely to require additional learning, not least because the tools for accomplishing that task and the environment in which the task must be accomplished will have changed. Computer programmers are educated and socialized to expect this, to embrace the endless struggle to acquire skills that will shortly become obsolete, and to mitigate its difficulty by working within a community of specialists: if one member of the team focuses on maintaining current database-design skills and another keeps up with the latest developments in user interface tools, the effort "to master all the new software programs coming out at an accelerating pace" becomes feasible by virtue of having been divided into smaller pieces. Otherwise, exhaustion and disappointment as expressed by Coover are natural responses to the realization that it is not possible, no matter how devotedly one works at it, to permanently master the entire moving target of computing technology.

For more than the briefest moment, computer-mediated hypertextuality cannot support an author's self-perception of solo activity in the way that paper-based book production methods do; I argue that this self-perception is false, created by a long tradition of obscuring the identities of those who create and process the raw materials of book production. Hypertextuality, though, comes from the tradition of software development, in which it is customary to acknowledge all contributors; software development's self-perception of collaboration is closer to that of film production, in which a cast and crew of hundreds may be identified by name and job assignment, than to that of book production, in which very few of those who worked to complete the project (an author, possibly an illustrator, perhaps an editor or translator) are ever identified. As David Bordwell and Kristen Thompson point out in *Film Art*, under the heading "Artistic Implications of the Production Process," working within a large collaborative group can be frustrating even where it is normative: "Big-budget filmmakers sometimes get tired of coordinating hundreds of staff and wrestling with million-dollar decisions, and they start to long for smaller projects that offer more time to reflect on what might work best" (25); neither Coover's longing to focus on "crafting words," nor his assumption that such focus is best achieved by reducing the complexity of the creative process, is his alone.

Bordwell and Thompson describe film production as "compromise within constraint" (25); I suggest that this is also an apt description of any hypertextual writing project, and the root of the frustration felt by authors who are surprised to find their expression of ideas limited by a mediating technology. Writing on paper is, of course, also limited by its mediating technology; the limitations may be unnoticed by an author who has never attempted to communicate complex ideas about movement or sound or size, but they are nonetheless there. Adapting to limitations, and continuously re-adapting as the limitations change, is another area in which those who succeed best in a cyberculture are those who can go beyond passive acceptance of change and actively seek and develop alternative techniques. Racing to catch up with externally-initiated change is frustrating; inventing internally-initiated change is exhilarating. Unless they are able to operate at the probably-unattainable level of designing and controlling all aspects of their own computing environments, rather than at the level of struggling to adapt to change controlled by others, authors who wish to produce solo digital texts may find this an impossible task. For authors who wish to work alone, dedicated to "crafting words" without being pushed and directed by ever-changing software tools, I argue that there are two viable options. One option is to accept hypertextual creativity as a normatively-collaborative act, and contribute their privately-crafted words (content) as one component of a larger hypertextual project in which other components (especially containers, such as the database and the user interface) are managed by other people; this preserves the project's hypertextuality but abandons the effort at non-collaboration. The other option is to abandon digital hypertextuality and its built-in expectations of collaboration and instability; using paper (or something resembling paper, including electronic paper) as a simple, stable, reliable container of words and ignoring any additional levels of creativity that might be made possible by the multiple levels of hypertextual construction.

Hypertextuality Requires Precision: The Limits of Simulation

It is easier for an author using raw material to independently create a unique paper text than a unique digital text for another reason, rooted in hypertextual ultrastructure rather than in psycho-social responses to change: digital expression requires precision. There may be infinite ways to create paper, using wood pulp or banana leaves or cotton rags or old newsprint or many other resources; the resulting paper may be smooth or rough, heavy or diaphanous, uniformly or irregularly colored, but it will still be recognizable and usable as a paper-like surface on which to place the marks that express ideas. The concept of "paper" exists as a broad continuum of physical forms; approximating any of those forms, or extending the continuum to accommodate new forms, is not problematic, as anything that resembles paper can be used as paper. In contrast, digital media are not in the business of accepting approximations and accommodating resemblances. An attempted digital representation of paper either is or is not a digital representation of paper, just as an attempted digital representation of the number ‘1’ either is or is not a digital representation of the number ‘1’; there is no possibility of a digital representation of “something sort of like the number ‘1’,” nor of “something that kind of seems like paper.” In digital operations, imprecision is error. A digital computer may seem to be endlessly patient, supporting multiple attempts to debug a system until it functions correctly, but what seems to be patience must not be interpreted as flexibility; exactitude is always required.

If a creative goal cannot be met while adhering to inflexible rules and operating within solid (but eternally evolving) structures, then fulfillment of that goal must be sought outside digital hypertextuality. With the ample precedent of poetic creativity, I argue that the requirements of hypertextual structure need not impose limits on imagination. Poets have found creativity within constraints more than possible, skillfully complying with the formal requirements of sonnet or haiku while “crafting words” in innovative ways. To find room for creativity within the digital structures of hypertextuality, writers need only develop two kinds of poetic skill: clean separation of their own free-form ideas from the fixed-form requirements of the structure, so that ideas transcend but do not damage (and are not damaged by) the structure; exploitation of structural features as if they were natural partners of the ideas expressed within the structure, so that awareness of the structure (requiring a moment of hesitation, a separation, a repetition) only reinforces understanding of ideas expressed within. Creativity of this sort can only benefit from collaboration between authors of content and designers of containers, enabled by the multi-level structure of digital hypertexts.

CHAPTER VI

"SAVE AS"—REPACKAGING, REPURPOSING, AND RESPONSIBILITY

Texts of all kinds can be used for purposes their creators did not anticipate; to say that digital texts are more flexible and adaptable than paper texts in no way illuminates the differences between digital and paper texts. In this chapter, I identify some of the sources of digital texts' adaptability to new uses, which I argue are based in their ultrastructural separation between container and contents. I also propose some terminology with which to describe the nature of the differences between a source text and an adaptation of that text to a new purpose.

In this chapter, I introduce the concept of *structurally non-identical copies*, in which content is preserved but structure is intentionally modified to adapt the copy to a use unlike that of the original. I also introduce the concept of *transitional copies*, created only for the short-term purpose of creating a bridge between one use (such as browsing on the Web) and another (such as reading on paper); using the process of printing from a digital text to a computer-controlled printer as an example, I argue that a key difference between digital and paper texts is their need for a supply of raw materials in advance of an adaptive re-use: printing requires blank sheets of paper, but there are no blank hypertexts, empty containers awaiting the imprint of their contents.

Another way in which digital texts differ from paper texts, I argue, is digital texts' potential to carry within themselves the means of their own adaptation to a new use. While a paper-like text cannot be altered without the assistance of an external mediator, a software-like text can be designed to serve as its own mediator, recognizing when an adaptation to a new use would be appropriate and proceeding to perform that adaptation. I also argue that digital texts of many kinds, not only hypertexts, can carry within themselves the means of recognizing and advising against improper use, of seeking assistance from human or software collaborators, and of identifying themselves as modifications rather than original texts; because a modification can be an act of imagination or of disinformation, texts which identify themselves as modified can identify themselves as fraudulent or otherwise invalid.

Digitization is widely known as the process of creating a digital alternative to a non-digital text; *mobilization* describes the related process of adapting a digital text designed for a large, desktop screen for use on a small, hand-held screen. Both of these processes operate on multiple levels, providing readers with access to the surface level of a new text which is supported by deeper structures. I argue that other adaptive processes similarly interact with textual structures: *citation* extracts a fragment from one textual container and injects it into another; *isolation* strips away the entire textual container, dividing a holistic text into its components which can then be re-assembled into a different container for a different purpose. For informational texts especially, I argue that digitization projects must not be seen as ends in themselves: further adaptive processing of a digitized text, isolating its components and placing them within the structure of a database, can make the text useful for active inquiry in addition to passive reading.

Adaptation: Contents Aware of Containers

A digital text can be perceived as damaged or useless when it is merely being examined through an ill-focused lens. For example, Figure 52 compares the same Web site, www.tamu.edu, as seen on a laptop computer screen and a handheld "mobile Web" telephone screen.

Figure 52: A Web page designed for mediation by a desktop computer is unusable on a mobile device.

When seen as a fragment of itself, occupying the entire small screen of the mobile device, the page appears to be useless, consisting only of colored blocks. The remainder of the page can be examined, fragment by fragment, by moving the mobile screen's slider or by selecting the "More" option, but the operator of the mobile device can do nothing to make the page better suited—fewer, smaller images; most-used links positioned highest—to the nature of the small screen: that intelligence either is or is not built into the Web page itself.

The possibility of a text being placed in a container for which it was not designed and in which it works badly is not unique to Web publications. What is unique is a digital text's potential to warn, as shown in Figure 53, when an attempted use is inappropriate.

Figure 53: A properly-prepared Web page can recognize inappropriate use and suggest alternatives.

For paper publication, the paper itself cannot warn that it may not be the most appropriate choice in context: load fluorescent yellow paper into a photocopier's input tray and press "Copy," and copies will be faithfully produced on that paper, whether or not this is the best available combination of contents and container. Bright yellow paper may be an excellent choice if the printed message is meant to attract shoppers to a sale, or it may be a poor choice if the printed message is a letter of application for employment at an insurance agency; only the human involved in the production process, not the paper nor the photocopier, can judge whether the container suits the contents. A Web page, though, can be equipped to detect that it is being used in an inappropriate way and suggest its own alternative: at www.mobiforge.com (MobiForge), code built into a page designed for large-screen desktop computers can detect the presence of a small-screen mobile device and suggest, as there is a separate version designed to suit such devices, that the visitor might be better served by that alternative; providing a hyperlink from the large-screen version at www.mobiforge.com to the small-screen alternative at mobiforge.mobi is only the final surface-level step in that process.

For two popular Web sites, www.google.com and www.wikipedia.org, the intelligence necessary to identify and adapt to the presence of a mobile device has been implemented and automated: rather than suggesting that a visitor might prefer an alternative version of the page, the page simply presents that screen-appropriate version. For both sites, Figure 54 compares the version designed for large desktop or laptop screens with the successfully mobile-enabled versions re-designed to suit small handheld screens.

Figure 54: Successful design for mobile devices brings most-used components to the strongest positions.

In both cases, the mobile version is a streamlined version of the desktop version, with non-essential features (such as Wikipedia's central image of a globe) eliminated or (like Google's options to restrict the search to images, maps, or news) relocated to less-prominent areas of the screen. In this, the design of an effective mobile Web page has much in common with the design of the front page of a printed newspaper, placing the most highly-valued components "above the fold," at the top of the small screen. In both cases, the top-heavy design is in part motivated financially: placing the most eye-catching headlines and photographs at the top of a newspaper's front page improves the page's usefulness as advertising to its own potential readers; placing the most useful components of a Web page at the top of a mobile screen reduces the time a reader must spend scrolling forward on the page, thereby potentially reducing charges for wireless connection time.

With current Web browser technology, the process that has come to be called "mobilization" can operate in any of the ways shown above, depending upon the methods implemented by a site's webmaster: automatically provide a mobile visitor with an appropriately-designed page; automatically suggest an appropriately-designed page but require the visitor to confirm the choice; make no effort to detect and adapt to a visitor's screen size. For the Web sites I manage in my own practice, the choice to do nothing remains in effect. Data collected by StatCounter shows that, in most weeks, less than one visitor attempts to read a page via a mobile device for any of the sites I manage. Such attempts are probably highly unsatisfactory, as can be demonstrated with mobile-screen emulation tools such as DotMobi's www.mtld.mobi/emulator.php, but mobilization remains a low priority for these sites because few visitors are directly affected: as the number of mobile visitors increases (as it seems bound to do, given the growing popularity of Web-enabled hand-held devices) and the labor-intensiveness of Web page redesign for mobilization decreases (as it is also bound to do, given that software becomes progressively easier to use and more feature-rich as it evolves), there will come a point at which it makes sense to directly address the needs of mobile visitors. The controlling effects of human choice and prioritization here are important considerations for all digital hypertexts, not only for Web pages: there may be things which a hypertext has the technical *potential* to do (for example, redirecting visitors to a more appropriate version of itself) but which it does not *actually* do for reasons which are not technical but human (for example, the webmaster's decision that the effort to implement this technique is not yet cost-justified). Bearing this human-induced variability in mind, observations that relate not to specific examples but to broad classes of hypertexts are most valid when most porous: "Hypertexts adapt to varying screen sizes" is problematic, because not all hypertexts have been given the tools to do this; "Hypertexts can be made to adapt to varying screen sizes" is more true, because it acknowledges the place of conscious human choice in constructing a hypertext to exploit its full potential.

Remediation: Creativity and Falsehood in Digital Images

In *Remediation: Understanding New Media*, Jay David Bolter and Richard Grusin explore and illustrate some of the ways in which our ability to digitally manipulate images blurs the categories in which we have customarily placed them. A digital photograph, a precise representation of the shapes and colors at which the camera was pointed, can easily be altered so as to appear imprecise: sharp edges can be softened, bright colors can be faded, crisp lines can be represented as charcoal rubbings or watercolor brushstrokes; what began as record of reality can become an expression of emotion, communicating an imagined scene rather than an actual one. For works of art, this ability to alter the original image may create a different kind of art, appreciated for different reasons. Creation of an altered photograph requires two kinds of skill: the photographer's ability to find and capture an image; the graphic artist's ability to adapt an image to a new purpose. The entire skillset need not reside in a single person, although the widening availability of low-cost, easy-to-use photographic editing software is making that increasingly feasible.

Remediation may be a perfectly legitimate artistic process, just another demonstration that artists are inspired by and build upon each other's efforts, and that they can use their own early works as bases for more-evolved later works. For works of information, though, the ease with which new forms can be created from old creates new levels of anxiety about not only the possibilities for technological and human error but about opportunities for uncontrollable proliferation of intentional falsehoods. Digital editing of photographic evidence has been a seething source of such uncertainty.

In a well-publicized recent example, photographs of a July 9, 2008 missile launch in Iran were used as evidence to support conflicting stories: perhaps four missiles were launched, or perhaps one missile was defective and only three were launched; both versions of the story were supported by photographs. Two days after the missile launch, as explained in an *ABC News* story (McGregor-Wood) citing an "expert in digital picture editing" and a "photo technician," the hoax was recognized by examining the structure of the four-missile image: in that image, one missile includes impossibly-identical elements of two other missiles in the group; the visual perspective is also impossible, showing two missiles at the same size but at different distances from the camera. Comments left by readers of the *ABC News* story at www.abcnews.go.com/International/story?id=5348664 point out that this kind of creativity—not transforming one kind of art into another kind of art, but changing an image representing reality into evidence supporting a lie—may be more pervasive and more consequential than has been appreciated. "UYOH1234" remarks "gee, won't we all feel silly when WWIII starts over a photo shop project"; "UA" follows up with "It can be done.I have corel photo editing and I could clone another missle in there easily.Kinda scary a skill even I have could conceivably be used [t]o start a war*DISCLAIMER*I did not do this one." Each comment mentions a popular commercially-available software tool that could be used to digitally alter a photographic image: Adobe markets a feature-rich version of Photoshop to "professional photographers, serious amateur photographers, graphic designers, web designers" for $649, and a simplified alternative (Photoshop Elements) for $80; the other photo-editing tool mentioned, from Corel, offers the "aspiring photographer" a "digital photography toolkit" with which to "get creative" for $69.99; competing products, including Google's Picasa, are also available at no cost. Images captured on photographic film can be intentionally distorted in the process of production, but the skills and equipment to do this credibly are rare and costly and of limited application: once the image is committed to paper, subtle alterations become impractical. For digital images, the ability to examine an image is not naturally separated from the ability to create and modify one's own copy of it. Copies can be created and altered with inexpensive, ubiquitous, easy-to-use tools, which are the same tools used to make the image more usable by changing its size or sharpening its focus:

without the need to commit a “finished” version of the image to paper for distribution, a digital image can be continually modified throughout its lifetime.

Structurally Non-Identical Copies: Creating a New Container by Moving into It

One of the ways in which digital texts differ from printed texts is the ease with which digital texts create other digital texts; that ease of creation is built into the software that constructs and modifies them. Paper is a passive, stable, dead-end material: the words on a piece of paper can be copied to another piece of paper, but that reproductive process requires mediation by an external agent such as a human scribe or an electronic photocopier. Computer-mediated digital documents, though, are made active by their mediating technology. The same software that makes it possible to read the text almost certainly makes it possible to copy it; whatever mechanism enables copying almost certainly also enables revision of the copy; the tool that creates a revision almost certainly imbeds within the text a record of its own activity. Where there is an interruption of this progression from reading an original to keeping a copy to modifying a copy to tracing the copy’s history of modifications, the interruption is a mark of human interference in the “natural” order by disabling a “normal” function.

Digital photographs routinely carry within themselves an indication of how they have been altered; Figure 55 shows two photographs, one the original and the other a modification, both of which I created.

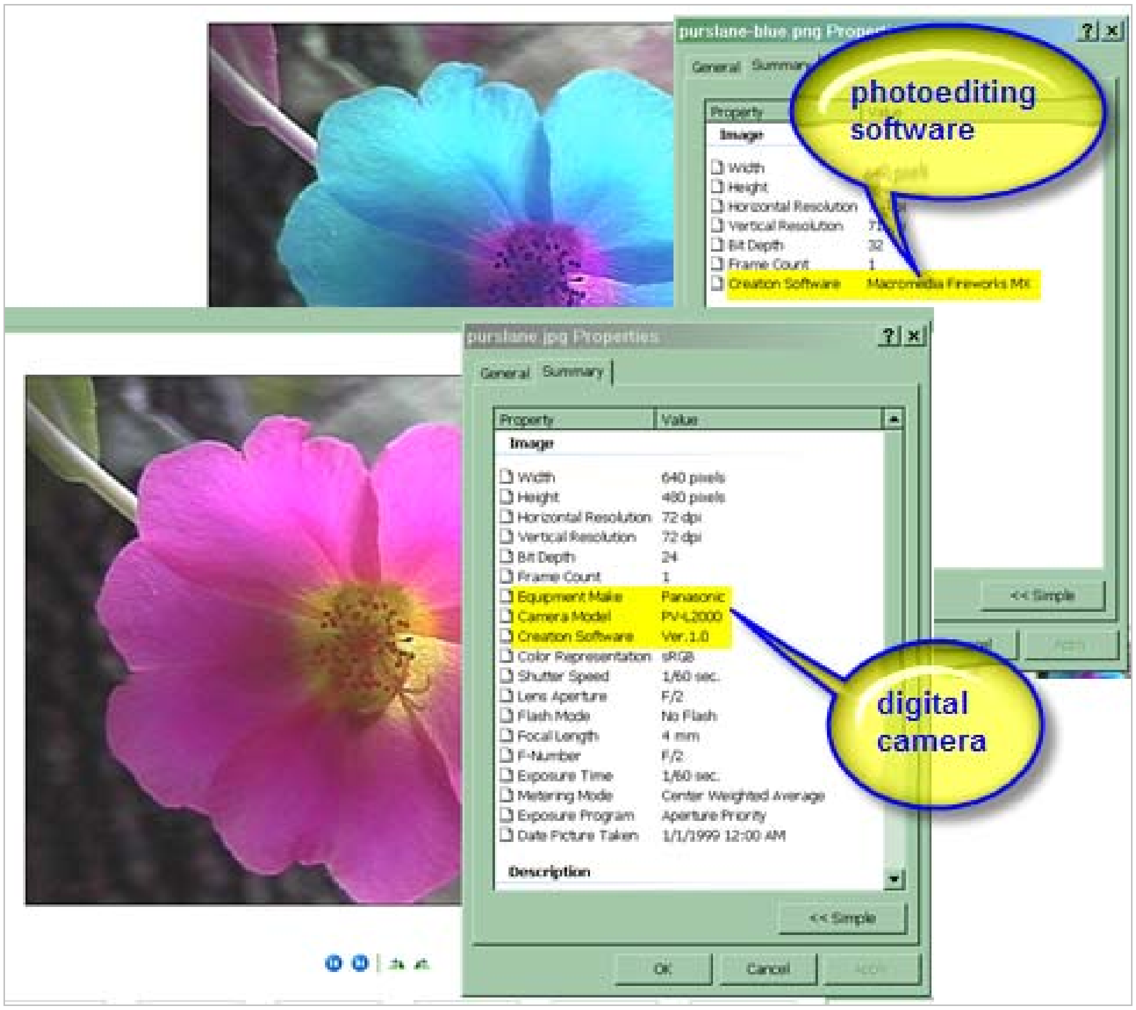


Figure 55: File properties identify one photograph as the original and the other as a modification.

The file properties shown beside each image are available without special-purpose software: the operating system (Windows XP Home Edition, in this example) provides built-in support, allowing the user to display basic facts that, with some understanding of the technological context, connect one of the photographs most closely to the natural world. Even without the content-specific awareness that, for the purslane flower pictured in Figure 55, one color combination (pink petals with yellow center) occurs in nature and the other (blue petals with pink center) is completely imaginary, the choice is obvious: the properties description stored internally with each photograph shows that the pink flower was created by a digital camera (a Panasonic PV-L2000) and the blue flower was created by photoediting software (Macromedia Fireworks MX); changing the normally-visible content of the image (shifting the flower's hue from pink to blue) also changes the description stored inside it which is accessible through the operating system's File—Properties tool.

Not all methods of creating and altering digital files leave such obvious evidence as the photo modifications discussed above, and not all the evidence than can be accessed is valid; the pink flower, for example, was photographed in August 2008 but, because the clock battery in the camera had been removed, the date stored with the image is January 1, 1999. This photo was taken in Texas, where purslane is not likely to flower in January, so this is a case in which contextual understanding can help in interpreting the validity of the evidence: information available from the source closest to the image itself (the camera, which records the image's size, resolution, shutter speed, focal length) is more likely to be valid than information collected from an associated-but-independent source a step removed from the image (the camera's clock, recording the image's timestamp). At an even greater distance from the reality captured in the image, human-crafted descriptive tagging such as img alt='pink flower' can be accidentally or intentionally deceptive: a Web browser can only display or read aloud the alternate text, with no means of judging how accurately it describes the image. For digital images, recognizing the separate contributions of multiple humans and image-capture hardware components is as necessary to valid analysis as recognizing the separate layers of textual construction is to understanding a hypertext: the camera's digital clock and digital lens operate independently, storing their results together in a digital image, just as a WordPress blog's theme and postings are managed independently but combined when a blog page is presented in a Web browser.

For non-photographic digital files such as word-processing documents, much of the automatically-generated descriptive information relates to time rather than, as in the photographs above, to production methods. Microsoft Word, for example, provides a Properties option on its File menu which displays four dates for a document: its creation, its most recently saved modification (writing), its most recent access (reading), and its most recent printing. Some statistics about the document are also reported: how many revisions have been saved, how many hours it has been open for editing (whether or not editing occurred), how many pages, paragraphs, words, and characters it contains. All this information relates to the current state of the document as a whole; information about the methods by which it was constructed, such as how much of its content was typed versus pasted in, or the ratio of words to images, or the extent to which formatting is controlled automatically by styles and macros, is not made available.

Human users cannot access a digital text, nor any self-descriptive information that might be stored with the text, without the mediating assistance of supporting technology: a browser, editor, reader, player, or some similar tool. Such tools, in addition to whatever simple processes of recording typed words they might have first been created to support, routinely support a variety of features for producing what I call "structurally non-identical copies" of the text. In a structurally non-identical copy, the linguistic content of the text is accurately transmitted but elements of its structure are changed to adapt it to a different use. For a database-management tool such as phpMyAdmin, this makes it possible to copy the contents (but not the structure) of the database to other output formats such as spreadsheets, word-processing documents, markup for typesetting and Web publication, and electronic paper. The same Export function of phpMyAdmin also supports creation of a structurally-*identical* copy, another database, but only indirectly: exporting the current database to its Structured Query Language (SQL) definitions creates statements which describe the contents and structure of the database; saving those statements and executing them, perhaps on a different computer or as a way of recovering from damage to the original computer, will cause a new database to be created. This is another case in which descriptive language—SQL code—creates reality—a database—in the process of describing it. The description that creates the database's reality, though, is not the noun form (the collection of descriptive statements) but the verbal (the process of executing the description).

In word-processing software, basic commands also copy text originally shaped for one purpose (private page-by-page reading on 8 ½" x 11" paper) into a container shaped for a different purpose (public slide-by-slide presentation projected onto an auditorium-sized giant screen, or a continuously scrolling Web page on a hand-held mobile screen). The same words, transferred from a private memo to a public presentation, become available to different readers in different contexts by virtue of their movement into new containers. The design of any text-mediating tool determines which kinds of new containers it will enable its users to create, and which containers can be directly created and occupied (File—Save As) and which require structural changes (File—Export) to adapt the contents to the new container.

In large part, these design choices express the toolmakers' understanding of the relationships between the current tool and its environment, which includes human users but also includes the tool's predecessors and competitors as well as the operating platforms on which it is expected to run. Users of a digital text may wish to create a new text from it, but can carry out that wish only by choosing among the options provided when the mediating software was created; when those options are inadequate or unobvious, the new text may be created by defective means (such as Copy—Paste rather than File—Export), causing it to be structurally (but perhaps not visibly) incompatible with its purpose. The "Namespace" section in Chapter IV, in which a properly-constructed iCalendar Web page becomes improperly-constructed electronic mail as illustrated in Figure 35, explores one example of this pervasive kind of error. Office productivity software such as word-processing tools is a ubiquitous source of structurally non-identical copies. Once created, a copy is not structurally connected to its original; the copy begins an independent life in its separate container, where its content can also be modified as appropriate to its new purpose.

One result of this ease of relocation is that physical labor is conserved and re-usable; once a text has been typed, it need never be typed again to make it usable for a different purpose, such as moving it from a spreadsheet (structured to support calculation) into the body of a proposal (structured to support private consideration) or into a visual presentation (structured to support public explanation). Another result is that appearances can be preserved; rather than setting the old words into new type, thus possibly introducing new errors, not only the text but formatting information such as font faces, sizes, and colors can be moved with the text, allowing it to retain its character as a shouting headline or a subtle footnote. Using software to create a new digital text from an old one is not like using a photocopier to reproduce a printed text onto black sheets of paper. A computer is not pre-loaded with blank documents of type .doc or .htm or .pdf, available to be filled with text in the way that the blank sheets of paper in a photocopier enter the machine empty and emerge from it filled. New digital texts are created of no raw materials other than the original digital text and the software's instructions as to how to create digital texts; there is one, and then there are more, but there never was a stockpile of unused potential digital texts.

Transition: Creating a Temporary Container

When a digital text is copied to a non-digital (paper) text through the software-controlled process of printing, the blank paper (container) *does* exist before it is filled, waiting to be imprinted with the text (contents) of the copy: creating the paper copy does not create the paper, but causes the paper to be used. The printing process also creates an intermediate container, in which the digital text as displayed on a computer-controlled screen is transformed into an alternative version of itself that can direct a computer-controlled printer. As shown in Figure 56, printing creates a new, normally-temporary, digital version of the document, in which normally-hidden ultrastructural formatting instructions are imbedded to make the digital text suitable for recording on paper, its final destination.

Figure 56: HTML (top) formats content for display by a Web browser; PCL (bottom) formats the same content for a printer.

Because the nature of the instructions used in formatting printed output may vary according to the manufacturer and model of the printer, "print" is no more a monolithic identity for computer-generated texts than is “digital.” A digital text can reside in a public Web page or in private electronic mail message or a variety of other document types, some but not all of which may be formatted as hypertext; copied to print, the text can be adapted to many sizes and thicknesses of paper, including adhesive label paper, glossy photographic paper, transparent plastic “paper” for presentation slides, and heat-sensitive T-shirt transfer paper. “Printing” a text can also mean transforming it into audio signals for transmission via fax, or preserving it as “electronic paper” in Portable Document Format. In the example shown in Figure 56 above, the printed copy is to be produced by a Hewlett-Packard LaserJet 3015 printer; instructions to that printer are provided in Hewlett-Packard’s Printer Control Language (PCL). Like the iron-on T-shirt transfer, the PCL file is meant to be destroyed in use; its value is as a momentary bridge between one relatively permanent form (a Web page) and another (a printed page).

For this illustration, I preserved the normally-temporary intermediate PCL text by instructing my system to print it to a file rather than to a printer; unlike HTML, most of the contents of this PCL file are not naturally readable as human language; because PCL files are not ordinarily displayed on computer screens, they are not encoded in ASCII or any other display-compatible symbol system. Elements of the PCL file which are easily readable identify the text to be printed on each page as an identifying header: “Please consider the environment before printing this email,” extracted from the <title> in the original Web page’s <head> section. Also clearly readable are the name of the computer from which the print request was submitted (“COPAL”) and the name of the user (“Rose”), logged in that computer, who submitted the request. While the data imbedded in a temporary PCL file is similar in nature to the data collected and stored by Web usage tracking tools such as StatCounter, its purpose is active communication rather than passive tracking: tying the print request to a human requestor who is logged in at an identifiable computer enables the printer to ask an interested human for assistance with physical activities that a printer cannot perform independently, such as clearing paper jams, adding ink, and replenishing the supply of blank paper. This is additional support for three of my earlier observations, which I now argue can apply to digital texts that are much simpler and shorter-lived than Web pages: computer systems and the texts they mediate are normatively, and sometimes necessarily, collaborative, with both natural and artificial intelligences eligible for membership in the collaborative community; software-like texts are constructed to provide their users with access to sources of help, even when the software's user is a machine and the machine's source of help is a human; because anonymity does not support the need for mutual assistance within the collaborative community, identity is recorded in deep structures even when it is not published.

Citation: Inserting Old Content Fragments into New Containers

New texts are routinely constructed of the fragments of old texts; extracted from their original contexts (containers), the old fragments can be arranged in a new container to suit a new purpose. The practice of including old fragments—quotations, footnotes, excerpts—in new texts is not new with digital texts, although some new methods of creating those connections—hyperlinks, pop-up windows—are only available in digital texts. This is Isaac Newton's idea of multi-generational collaboration, seeing far by standing on the shoulders of giants; the absent and the dead can legitimately be summoned (cited) to contribute their wisdom in aid of present projects. To *cite* (or *excite* or *incite*) an older text within and in support of a newer text in this way does not directly damage the older text, although it can provide readers of the newer text with invalid perceptions about the older text.

For example, in reading about software visualization in general and Kiviat (spider) diagrams in particular, I encountered Philip J. Kiviat's 1991 "Simulation, Technology, and the Decision Process" essay published in *Transactions on Modeling and Computer Simulation*; the essay is Kiviat's adaptation of his keynote address to an Association for Computing Machinery meeting of fellow specialists in computer simulation. In the address which became the essay, Kiviat credits then-Senator Albert Gore in 1990 with clearly communicating the essence and importance of simulation:

> It is hard to understand an ocean because it is too big. It is hard to understand a molecule because it is too small. It is hard to understand nuclear physics because it is too fast. It is hard to understand the greenhouse effect because it is too slow. Supercomputers break these barriers to understanding. They, in effect, shrink oceans, zoom in on molecules, slow down physics, and fastforward climates. Clearly, a scientist who can see natural phenomena at the right size and the right speed learns more than one who is faced with a blur (91).

These words, which I acquired from Kiviat and which Kiviat, having used them in two contexts, ascribes to Gore, are a small fragment of a July 1990 *Washington Post* essay by

Gore: in “Communications; Networking the Future; We Need a National ‘Superhighway’ for Computer Information,” Gore argues for government funding of the telecommunications infrastructure that would eventually be developed privately. While Kiviat credits Gore, Gore attributes the words to “Sheryl Handler of Thinking Machines Corp” who “testified recently before a Senate subcommittee” (B3). Gore’s citation of Handler in the *Washington Post* is not identical to the official transcript of her Senate testimony: among other minor differences, the official transcript of Handler’s statement is a record of speech rather than writing, interrupted by questions from members of the Senate subcommittee, including Gore (*Hearings*).

Handler’s, Gore’s, and Kiviat’s versions of the cited text, in the contexts each author created for it, are available in separate digital archives maintained by the U.S. Government Printing Office (for Handler’s testimony), the *Washington Post* (for Gore’s essay) and the Association for Computing Machinery (for Kiviat’s essay). Comparing the three sources points out that the cited fragment, while transcribed almost perfectly, is *used* differently in each: Handler seeks government funding for development of supercomputers, required by scientists but too costly to be profitable for private manufacturers; Gore argues that the government must support a telecommunications infrastructure for the same reasons the government supported development of the interstate highway system; Kiviat asserts to his colleagues that their shared area of specialization is “getting more exposure and is being recognized as a necessary way of doing business in an increasingly complex world” (Kiviat 91). By extracting the words from an older container and placing them in a newer one, multiple writers preserved the words but abandoned the previous author's purpose, possibly also shaping the cited authority's reputation in the new context: Kiviat enhances Gore's reputation as a politician sympathetic to technological causes, while obscuring Handler's reputation as a technologist who communicates well to politicians.

If all the source texts in this chain of citation were hyperlinked to each other, the true source of the words (not Gore but Handler, or some unnamed staff writer at Thinking Machines who prepared Handler's Senate testimony) might be more easily discovered. Three factors impede establishment of such a connection. First, the source texts are all stored in archives managed by independent organizations who may prefer to see themselves as self-sufficient endpoints rather than as steps along the path to some externally point of origin; whatever inter-archive connections are possible may not be seen as beneficial to the human caretakers who would be required to implement them. Second, the archived digital versions of these source texts are paper-like, with their current forms lacking the ultrastructure to hyperlink a few lines in the citing text to the source of those lines in a cited text. This can be overcome by various means, all of which amount to providing each page with an ultrastructure to enable hyperlinking: a hyperlinked image map could be added as an overlay to each scanned page image, so users who click on a portion of the image are directed to a relevant Web page; the content of the scanned page images could be transcribed, manually or by Optical Character Recognition, creating software-like alternative versions in which text-as-text rather than text-in-image can support the relevant hyperlinks. The third obstacle is more profound: even with hyperlinks in place, current technology does not provide for multistep hyperlinking, from a new text through a series of intermediate texts to the original source of a citation: I envision some future implementation of what I call *genealogical hyperlinking*, in which cited text could be permanently connected through a series of intermediaries to its origin, with the entire hyperlinked series visualized for readers as beads on a string and any of the beads selectable as the destination of the hyperlink; as currently implemented, hyperlinks create paths to, not through, pages, so hyperlinking remains a matter of following a path to one page and then seeking that page's path to another page. Genealogical hyperlinking is akin to Theodor Holm Nelson's "xanalogical structure," a proposed system of overlays that would create "visualizable re-use" (2) among hypertexts; it also resembles current practices for citing indirect sources in written texts (Gibaldi 245) and oral traditions, in which it is usual and valid to acknowledge an idea as having come from X who heard it from Y who learned it from the inventor, Z. With continued development, citation in digital hypertexts may come to work as smoothly as citation in oral traditions.

Isolation: Decapsulating Pages, Stories, Data

The process of extracting contents from their container can strip them of more than their connection to an earlier author's intentions; while *citation* carries only a fragment of an older text into a newer one, the process which I call *isolation* deliberately separates the entire text from its bindings, whether those bindings are logical or physical. In pulling all the contents out of their container, damage can result. Physical damage may be unintentional and indirect: as libraries and businesses scan old paper documents into new digital images, the paper itself loses what perceived value it had as a container of information and may be discarded as a space-saving measure. The destruction of paper originals can also be a conscious act, a price willingly paid for the creation of their digital counterparts. For example, while the mission of Project Gutenberg, “to encourage the creation and distribution of eBooks,” might encompass new books which have never existed in any but digital form, examination of their Web site at www.gutenberg.org makes it clear that, as practiced, the mission is to create digital books from printed books, even if that process destroys the printed book. Instructions to Project Gutenberg volunteers at www.gutenberg.org/wiki/Gutenberg:Scanning_FAQ compare the relative merits of several methods of accomplishing the transformation: cutting pages from a printed book’s binding is highly recommended, as this makes it possible to obtain a high-quality, high-speed optical scan through an automatic document feeder; preserving the book by leaving the pages anchored within their binding allows only a lower-quality, lower-speed scan on a flatbed or scanner.

The decapsulating process of isolation can separate what was a unified text into a loose collection of separate components: in Project Gutenberg’s example of a book cut away from its cover for digitization, the components are *pages*; other kinds of texts might be isolated into other kinds of components in the process of digitization. For example, comparing the front-page contents of the printed edition and the digital archive for the “same” issue of a newspaper shows that the isolated components of the newspaper page are *stories*, as illustrated in Figure 57.

Figure 57: Order in the *New York Times* digital archive (L) is unlike the print edition (R).

For the May 8, 2007 *New York Times*, the archived front-page stories are the same as those on the printed front page; however, separated from the order and structure within which the print editor placed the stories, important contextual information is absent. Steven R. Weisman's story about the World Bank, for example, is given a high-priority position on the printed page, with that priority reinforced by a heavy, all-capitals headline; in the archive's list of stories from the front page, it is simply the seventh of seven stories, positioned last on the list because the other stories on the page were filed later than it was. The print edition's structure, rich in illustrations as well as visual cues as to stories' relative priorities, offers a holistic impression of the events and ideas of the day; however, the archive better supports investigations of how a topic was developed over multiple days. Unlike mobilization tools, which possess the intelligence to detect an inappropriate use and propose an alternative, archive retrieval tools don't yet recognize that a visitor seems to be interested in the events of a single day and suggest that examining the digitized image of the day's front page might be a more useful approach than reading the isolated stories that were published on that page.

Components isolated from larger texts can be stories or pages; they can also be much smaller fragments, the basic facts themselves, re-ordered and re-presented to serve a new purpose in a new text. In the case of the *New York Times* archive discussed above, isolating front-page stories from their container caused a loss of meaningful contextual information. At the same time, pulling components out of a rigid structure in which they are contained can free them to create new levels of meaning: I argue that, for work that has already been digitized, this is the next logical step in expanding the usefulness of a digital text. To illuminate this process, the following case study compares two late nineteenth-century informational texts, both available in digital archives, which address the same subject matter—an annual summary of lynchings in the United States—by presenting the same data within different structures.

Case Study: Ida B. Wells and Lynching Statistics

The December 31, 1893 *Chicago Tribune* marked the end of that year by devoting a portion of the year's final issue "to take a retrospect of the old year, to show what has been accomplished, whether the world has progressed or retrograded in certain directions—in short, to give a comprehensive record of the year's doings" (20). In "The Record of 1893," statistics in "the moral departments" for 1893 are assessed as having somewhat improved from 1892, with murders and lynchings down, although suicides and executions are up:

> As the outcome of murders and other crimes 126 persons have been executed-legally, as against 107 in 1892, and 200 have been lynched, as against 236 in 1892. The increase in legal and the decrease in illegal hangings would indicate healthier conditions in the operations of justice, for it is the first time in fifteen years that the record of lynching has shown a decrease. The statistics, as usual, point to the South as the favorite locality of Judge Lynch and mob law. While 17 have been lynched in the Northern 183 have been lynched in the Southern States, of these 183 no less than 154 were colored men. Notwithstanding these suggestive comparisons there is room for hope that respect for law is increasing and that the popular sentiment of the country, joined to the efforts of some of the Southern Governors in securing legislation and penalties against mob law, is beginning to have some influence (20).

A few pages later, “Judge Lynch’s Work” presents lynching statistics detail: for every month of the year, all reported lynchings in the U.S. are listed by the date, the victim’s name and race, the crime of which the victim was accused, and the location of the lynching. In Figure 58, I have placed the September portion of “Judge Lynch’s Work,” which I cropped from the image of that page in the *Tribune*’s digital archive, beside my own visualization of the *Tribune*’s chronologically-based hierarchical structure in that portion of the report.

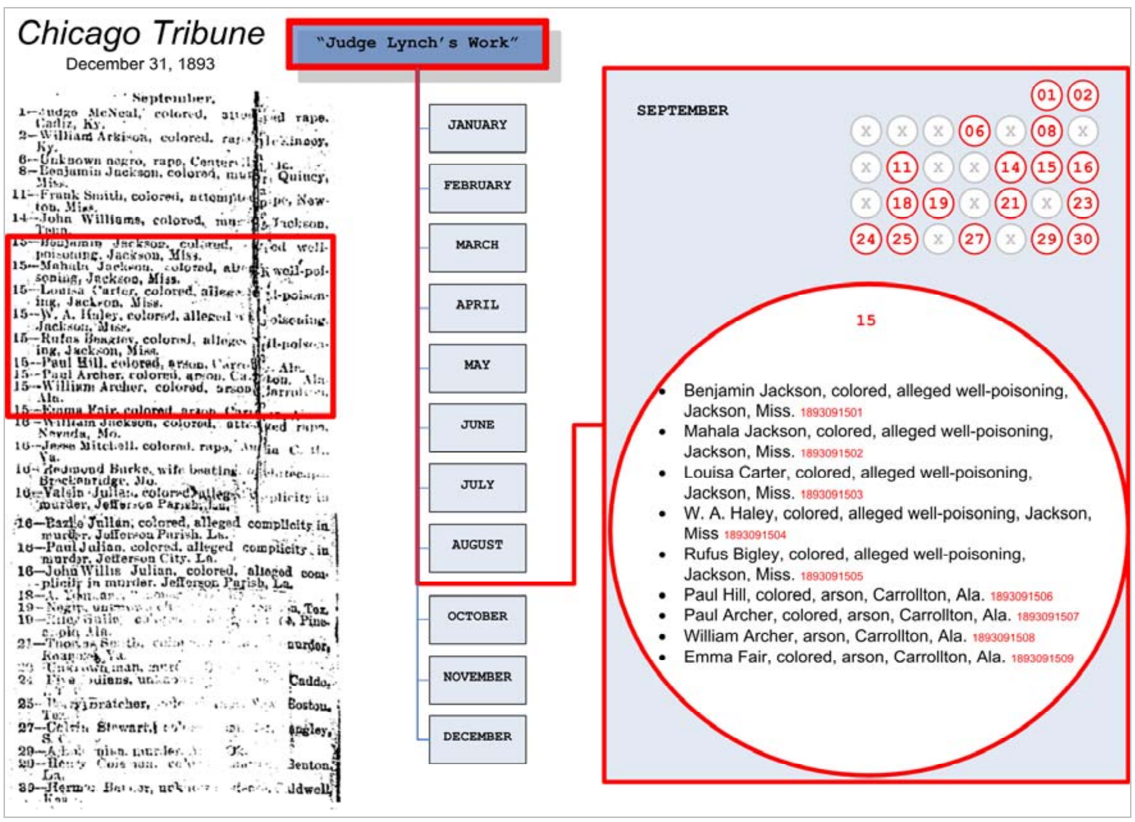


Figure 58: *Chicago Tribune*'s report of lynchings is organized chronologically.

The *Tribune* report is subdivided by month; within each month, lynchings are listed by date. When multiple lynchings are listed on the same date, their order does not appear to be meaningful: they are not alphabetized by the victims' first or last names nor by the locations of the lynchings. Because the *Tribune* organized its report through the lens of chronology, I designed my visualization to use the same focus, drilling down to a single month and then a single day. I expanded the month of September, listing all thirty days of the month and numbering in red the days on which the *Tribune* reports lynchings (such as the 1st, 2nd, and 6th of the month); days on which no lynching was reported (such as the 3rd, 4th, and 5th of the month) I have marked with an unnumbered grey "X." September 1, 1893 was a Friday; matching the dates with the days of the week, it is clear that in September 1893 one was least likely to be lynched on a Sunday. The one Sunday on which a lynching was reported, the 24th, was the second day of a three-day series of lynchings ending on the 25th; the date of the full moon that month; if one's purpose in reading "Judge Lynch's Work" were to learn how lynching activity relates to chronological events such as weekend holidays and lunar phases, the *Tribune*'s structure is ideal.

For other purposes, the chronological structure is less helpful. For the same month, the greatest number of lynchings for a single day was nine, on the 15th. In my visualization, I expanded the details for September 15th to include the data describing each of the day's nine lynchings. The nine victims seem to be associated with only two events: five victims in Jackson, Mississippi were lynched on the excuse of "alleged well-poisoning"; on the same day, four other victims in Carrollton, Alabama were lynched for "arson." Whether or not other lynchings occurred in the same places or for the same excuses in 1893 can be determined by examining the full listing of lynchings for the year; the data is there, but the structure makes obvious only the answers to questions about when lynchings happened, not why or where.

An alternative structure, suitable for addressing other questions, was developed by Ida B. Wells (1862-1931) and published under the heading "Lynch Law Statistics" in her 1895 anti-lynching pamphlet *A Red Record*. Within her own structure, Wells re-used the *Tribune*'s data, relying on the newspaper's credibility to support her own:

> For a number of years the Chicago Tribune, admittedly one of the leading journals of America, has made a speciality of the compilation of statistics touching upon lynching. The data compiled by that journal and published to the world January 1st, 1894, up to the present time has not been disputed. In order to be safe from the charge of exaggeration, the incidents hereinafter reported have been confined to those vouched for by the Tribune (69).

In Figure 59, I have included a portion of Wells' "Lynch Law Statistics" beside my own visualization of her structure. By preserving the *Tribune*'s data but placing them within her own structure, she transformed the *Tribune*'s annual report from a simple list of events into a collection of evidence supporting her contention that, contrary to then-current opinion, most lynchings had little to do with threats to white women. This is not change-as-error but intentional structural change as the same words move from one container (a general-interest newspaper report) into another (a special-interest anti-lynching pamphlet) to create a structurally non-identical copy.

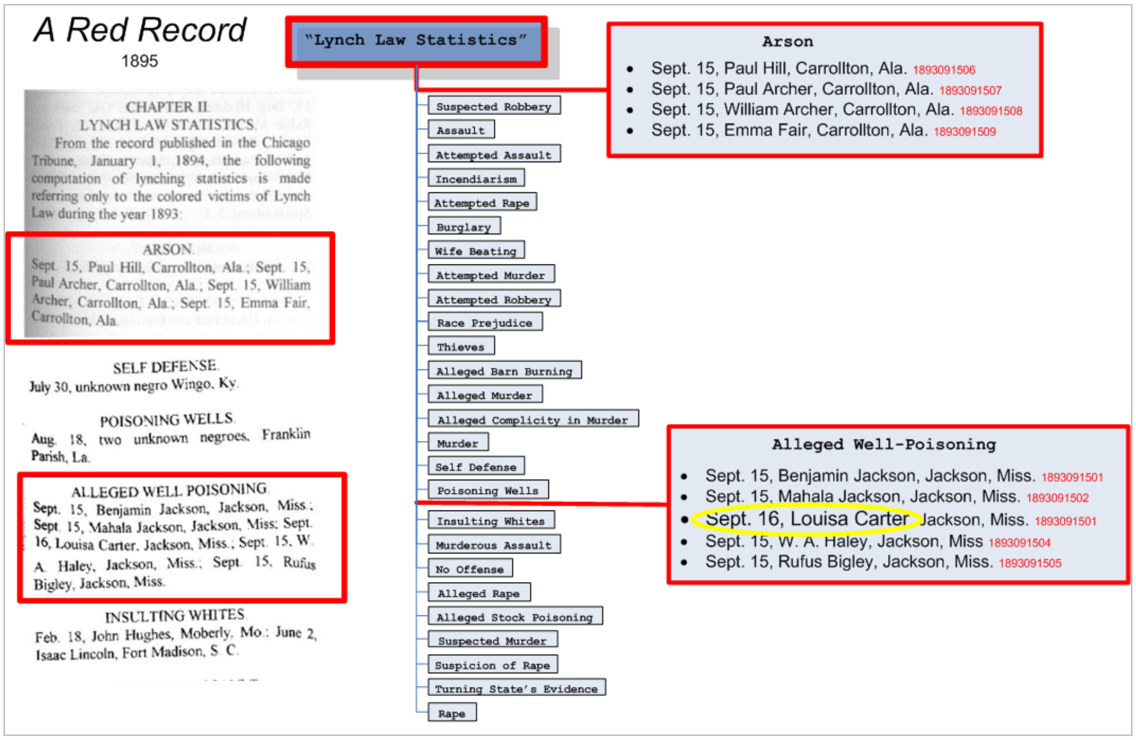


Figure 59: Wells' report of lynchings categorizes accusations made against lynching victims.

Wells' report is not ordered by date, but the events of September 15th noted in the *Tribune*'s text can also be seen here, grouped by the accusations against the lynching victims: "arson" and "alleged well poisoning," with no other examples of those accusations on other dates or in other places.

Hand-drawn visualizations such as Figure 58 and Figure 59 above, comparing the dissimilar structures within which two publications present the same data, can be useful as an initial approach to understanding the nature of an intentional structural change. Because it requires close reading, it can also point out dissimilarities in the data which may or may not be errors in transmitting the text. For example, the *Tribune* lists Louisa Carter as having been murdered on September 15, while Wells lists her death on September 16: perhaps this is an error on Wells' part, showing that she or a member of her staff garbled some of the *Tribune*'s data, or perhaps Wells had better information than the *Tribune*, such as the true date of Louisa Carter's death after she was attacked on the 15th.

Even for this small set of data describing 200 murders in 1893, fully visualizing both Wells' and the *Tribune*'s structures in a way that makes clear all their differences is not a feasible task for paper-based presentation. Also, even with digital drawing tools, the process of visualizing textual structure is tedious, time-consuming, and error-prone, just as Wells' effort to re-order the *Tribune*'s data must have been. Rather than manually creating larger visualizations or additional alternatively-ordered listings of events, future interrogation of this data could be well supported by moving the data into yet another structure, a digital database: I argue that this is the next logical step after digitization, so that a text can be not only read but analyzed. Placing the data in a structured database can make all its elements simultaneously accessible to direct inquiry: database queries can produce full lists in any order (SORT BY CITY WITHIN STATE) and extract portions of the list to support narrow inquiries (SELECT WHERE LASTNAME == "Jackson"); they can also feed automated visualization tools which, by exploiting sizes and dimensions unavailable on paper, can produce full and interrogateable digital models of the structures created by the *Tribune* and by Wells. Stored in a database, structured for active inquiry rather than passive reading, the statistics which the *Tribune* used to answer one question (When were people lynched?) and Wells painstakingly re-organized to answer another (Why were people lynched?) could be put to work answering other questions, posed by current readers rather than by Wells and the unnamed writer at the *Tribune*. The results of such inquiries can create new writing from the material that so energized Wells; for other texts, applying the same process of separating data from the original structure to re-store it in a database, can have similar benefits.

Integration: Contents, Containers, Contexts

Ida B. Wells is not the only pre-hypertextual thinker to consider whether placing contents into a container for which they were not intended might produce some differently-useful results. In the *Mishnah*, edited in about 200CE from earlier oral sources, Jewish tradition includes valuable insight into the power of containers to influence contents: as used in the Temple, utensils for liquids can sanctify liquids but not drystuffs; utensils for drystuffs can sanctify drystuffs but not liquids; none of these potentially-sanctifying utensils have that power if badly damaged or if used outside the sanctuary (Neusner 721, Zebahim 9:7 in Holy Things). The nature of a contained object *can* be changed, but only by a well-chosen container, used under the correct circumstances. The incorrect container, or the potentially-correct container in the incorrect location, has no ability to change its contents: the contents leak out and are lost, or they are simply unaffected by the contact. I think this observation, that contents can be profoundly altered by their containers, but only under narrowly-defined circumstances, is a useful way to consider the changes that are possible as a work moves from other media into hypertext, especially when a digital container does more than make a digital copy; by changing the structure in which words and images are contained, a digital container can change the ways in which they can be used.

CHAPTER VII

CONCLUSION

In the preceding chapters, focusing primarily but not exclusively on Web-based digital hypertexts, I have examined some of the ways in which texts change accidentally ("Chapter II: Click Here to Agree") and intentionally ("Chapter III: README"), explored a webmaster's access to the details of textual construction and use ("Chapter IV: View Page Source"), proposed and demonstrated several models of collaborative textual construction ("Chapter V: Edit This Page") and identified some of the processes by which texts constructed for one purpose can be re-constructed to serve another ("Chapter VI: Save As"). Throughout, I have argued that digital hypertexts differ essentially from paper texts in that hypertexts are constructed in multiple layers, with surface-level appearance and behavior, the text's content, controlled by sub-surface ultrastructure, the text's container(s). I have further argued that most readers, restricted to surface-level interaction with texts, have little access to the deep structure of any hypertext, and that such access is best obtained by collaborating with the person responsible for the hypertext's construction and care; I have supported this argument with examples taken from my own practice as a webmaster.

The study of hypertextual ultrastructure is complicated by factors in addition to the special privileges required for full access: multiple layers of containment and construction can be too complex and too rapidly-evolving to support direct observation and analysis. Other fields such as biology and chemistry, in which the object of study is difficult to observe and analysis requires simultaneous consideration of multiple unstable variables, have devoted a great deal of energy to developing visualization and modeling techniques to support their investigative efforts: I have argued for the necessity of developing similar methods to support hypertextual studies, supporting my argument with demonstrations of several methods of visualizing textual structure. In this, I am also arguing in support of two seemingly-contradictory statements about the relationship between visualization and critical thought. In "Theorizing Visual Intelligence," Rick Williams identifies visual intelligence as "the primary intuitive intelligence," pre-rational rather than irrational, so that "the initial, primary response to visual cognition is preconscious" (35): ideas expressed visually can be grasped rapidly and holistically. In *The Human Interface*, Jef Raskin urges programmers to push themselves to work in the opposite direction, verbally explaining their algorithms rather than relying on traditional visual development methods such as flowcharting, because non-verbal imagination relies on the cognitive unconscious, which "can sustain contradictions" as it does in dreams:

> to write out your intentions clearly in natural language forces you to make your reasoning process conscious, and it is in the cognitive conscious that contradictions most readily become apparent [...] writing comments makes you think through the problem an extra time, in a different medium, and from a different point of view" (194).

I think the proper combination of verbal and visual techniques, explaining images and visualizing words, will make it possible to study and understand the multi-layered nature of hypertextual construction and operation.

Findings and Recommendations

In this dissertation, I have argued that digital hypertexts have beyond-text attributes, created in a software-controlled layer beneath the surface layer of their contents, and that these attributes distinguish digital hypertexts from other kinds of texts. I have identified several of the *hyper*- attributes of hypertexts, including hyper-versioning, hyper-description, hyper-collaboration, hyper-identification, hyper-narration, hyper-control, hyper-linking, and the structural possibilities for hyper-publication and hyper-loss. I have argued that hypertexts cannot be usefully criticized without direct and thorough study of the ultrastructures that enable their *hyper*- attributes, and suggested some ways in which, by focusing on hypertextual structure and function at an appropriately granular level, a formalist criticism of hypertext can be developed.

By building on these ideas, thinkers in a variety of disciplines may find new ways to address specific challenges created by the growing pervasiveness of hypertexts in their own fields of interest:

- Textual scholars, interested in identifying the effects of multiple rounds of writing, editing, and publication on the development of a text, may benefit from observation of rapidly-evolving collaborative texts such as wikis in some of the same ways that biologists benefit from observation of short-lived animals such as mice. Humans and mice, and wikis and books, resemble each other in some basic ways but differ in enough details that any specific findings would require careful confirmation before being generalized from one species to the other. However, in both cases, a primary benefit of observing the faster-moving species (mice or wikis) is access to multiple generations of development within a relatively-brief timeframe: individual humans cannot similarly observe multiple full lifecycles of creatures, such as tortoises and religious texts, which typically live longer than we do. Observing many generations of rapid development of any text suggests evolutionary patterns: observing large numbers of errors supports organizing those errors by their causes and into categories; observing extended power struggles between collaborators (what wiki authors call "revert wars") ties some textual change to organizational and political change; observing the evolving text's responses to changes in its mediating technologies tracks the extent to which it is being enhanced (indicating expectation of future value) or maintained (indicating respect for its past value). For students of textual development, observations and experiments with digital hypertexts can develop and hone descriptive and investigative skills applicable to many other kinds of texts.

- Technical communicators may develop approaches of explaining and teaching that do not rely on the traditional method of asking users to match their actions to illustrated screenshots; as the surface-level appearance of a system becomes increasingly customizable to suit each user, the general usefulness of any set of instructions that relies on appearances decreases. In this, the concerns about Web page design expressed by the Americans with Disabilities Act are broadened from special-interest to general concerns: not only physical disabilities, but personal and technological preferences, cause users to interact with computers in unique ways; the "normal" user, operating like a large number of other "normal" users, is becoming a creature of the past. For digital instructions such as online help files, an alternative to presenting falsely "normal" illustrations is to detect the user's personalization settings and select illustrations and examples to match those settings; dynamic Web page design makes progress in this direction possible. For printed instructions such as installation manuals, an alternative is to simply abandon attempts to explain actions based on appearances and rely on each reader to match a desired action with the tools available in their personalized interface. Rather than "Click on the green button at the bottom of the page" (which, for some users, may instead be a blue hyperlink on the left side of the page), a useful instruction could identify the function to be performed: "Test the connection" sets the expectation that there will be a way to perform this action, and trusts each user to recognize what that way is in his or her personal situation.

- Rhetoricians, experts in persuasive language, may consider applying theirs skills to the growing need for effective *dissuasive* language on the Web. Hyperlinking and search engine optimization, implemented below the surface-level content, are highly effective in creating connections and attracting readers to Web pages; domain names, copyright claims, user agreements, warnings of adult content, and statements of acceptable behavior, all presented at the surface level, are largely ineffective in creating boundaries around Web pages and pushing inappropriate activity outside those boundaries. Tools to increase cooperation oppose tools to increase privacy. In a technologically-mediated system, changing the tools can relocate the conflict but cannot resolve it; it may be in the realm of human persuasion to develop methods of pushing the Web's readers and writers away from what *can* be done and toward what *should* be done.

- Professional journalists, competing increasingly (and often unsuccessfully) with amateur bloggers and other unedited voices, may devote additional energy to making their Web-published stories findable as answers to specific questions (keyword searches) rather than as announcements that might be discovered by browsing through all the pages of the "paper." Commercially-competitive Web sites accomplish this by using sub-surface markup to enlist the aid of search engine robots and spiders. For example, Daragahi and Mostaghim's June 15, 2009 *Los Angeles Times* story titled "Iranian protesters contest Mahmoud Ahmadinejad's reelection" marks itself internally with only two search terms: <meta name="Keywords" content="world, news" />; optimized to describe the story to search engines and through them to potential readers, the list of search terms could be much more detailed, built from the nouns in the story and offering it to those interested in the specific subject matter rather than as one of many items in the broad category of world news: <meta name="Keywords" content="Iran, Tehran, election, election results, reelection, protest, Mahmoud Ahmadinejad, Mir-Hossein Mousavi, opposition, censorship, voter fraud, Freedom Square, dictator, civil disobedience">.
- Linguists, especially those working in critical discourse analysis and critical cybercultural studies, can extend their work to address multi-level issues of anonymity and control in hypertexts. "Flaming," hostile linguistic behavior by anonymous posters in Web-based discussion forums, is a widely-reported phenomenon; underlying and enabling that behavior are the choices of webmasters

and moderators who, by the design of the forum, are provided with the means to interfere with public hostility but choose not to do so.

- Literary critics can apply formal, descriptive vocabulary to the multiple levels of hypertextual works of literature such as those collected at www.eliterature.org and typically described with high-level labels such as "Long fiction in English. Hypertext." Many kinds of hypertext exist; more granular identification enables better reader decisions about which items in the collection to pursue. Beyond initial categorizations, rich descriptions allow better critical thinking about how items in the collection relate to each other: two works of wiki fiction might be reasonably examined together, just as two gothic novels might be; comparing a set of static Web pages to a multi-user game might make as little sense as comparing a haiku and an epic.
- Librarians, archivists, and other textual caretakers, faced with the possibility of hyper-loss of valuable hypertexts, may adopt the same passion for business continuity and disaster recovery planning that characterizes data-centered commercial enterprises. In addition to in-house procedures to protect data and work around local instabilities such as loss of electrical power or network connectivity, large-scale collaborative efforts can provide offsite backups and mirror sites so that global access to hypertext can survive local loss.

By arguing for an ultrastructural understanding of hypertext as a basis for a hypertextual criticism which addresses the sub-surface differences between hypertexts and other texts, I am only partly advocating that understanding as an end in itself:

attending to the details of a text's internal construction, whatever those details are, is a practice that can be extended as new kinds of texts are developed. Hypertext is an important medium, likely to grow more important for some years before it is replaced or, like speech and writing and printing, augmented by other communication tools. Writing is a logical extension of the power of speech, as printing extends the range of writing and digital publication adds flexibility to printed publication; all these things are the bases of hypertext, but I do not know what next logical step will be built upon hypertext. Even as I describe texts in this dissertation as "digital," because the texts are mediated by computers and most computers are currently digital, I know that "digital" is itself an unnecessarily-limiting term: there have been non-digital analog computers, and there will be non-digital quantum computers; whatever forms of documents are mediated by future technologies, it is reasonable to expect that it may someday be archaic to call them "digital," although we may continue to do so in the same way we persist in referring to non-paper "pages."

Categories: Hypertexts Are Texts

Because hypertexts are texts, it is completely logical to ask the same questions of a digital hypertext that might be asked of a text recorded on paper or film or any other medium: How was this created, and by whom, and for what purpose? How is the text I am now experiencing different from what its author imagined that I might someday experience? How is this example typical of, or unique within, a category of similar texts? For hypertexts, this last question of relationship to a category or *genre* has often been poorly answered: as "paper" is not a genre, neither are "digital" and "hypertext."

Digital texts exist in a wide range of forms, many of which are hypertexts; other digital texts function as electronic paper, digitally stored but functionally flat in the way that paper itself is physically flat, lacking the ultrastructure necessary to do more than publish their surface-level contents. Digital texts which are hypertexts can have little in common other than that fact; in this dissertation, I have drawn examples from some of the most widely-used forms of hypertext, including electronic mail messages, word processing documents, and static and dynamic Web pages; dynamic Web pages, constructed on demand to suit each reader, are widely used to create wikis, blogs, forums, shopping carts, and other containers of writerly reading experiences. At their lowest levels, these containers are largely similar: saved components retrieved from a database are combined with new components collected from a writing reader at a screen; the combination shapes the content of a new custom-made screen (container), from which the assembly process continues.

Nothing in the nature of a hypertext limits the genre of its content: a communally-constructed wiki page can contain an informational encyclopedia entry or a fictional short story. Hypertextual publications can be works of poetry or fiction or drama or journalism, recognizable as such by the same content-based criteria that would apply to a paper publication. Hypertextuality adds layers of potential complexity and expands the range of possible interactions between reader, writer, and text; a *hyper*-poem may be adaptive to its environment, providing each reader with a unique experience and opportunities to collaborate with other readers; however, it remains in its essence a poem.

Hypertexts are texts, and some elements of hypertextual construction resemble elements of other texts and are amenable to study in some of the same ways. Hypertexts are like drama, with computer languages encoding back-stage direction to control on-stage performance of human language. Hypertexts are like poetry, with structure and repetition creating a framework within which infinite variation is possible. Hypertexts are like fiction, with a narrator's voice guiding the reader to interpret events and consider alternatives. Hypertexts are like film, assembled by large teams of collaborators with complementary artistic and technical skills. Hypertexts are like software, most valued when most unstable: a "stabilized" software version is one that has been abandoned by its developers and will no longer be improved; as Diehl observes in *Software Visualization*, "all successful software gets changed" (129). Hypertexts are like speech, focused on action (enter, select, open, download, enable, submit, click, exit) and "encased in communal reaction" (Ong 46). Because they are *hyper*-texts, they can be all of these things and more, simultaneously, with multiple layers of structure creating multiple forms of interaction and adaptation.

Processes: Hypertexts Are Mediated

Because hypertexts are mediated by computer software, they are subject to the endless cycle of computer processing: input, process, output; output of one process becomes input to the next; the cycle covers the same ground, circling, or moves ahead to new ground, spiraling. Computer software operates on the basis of *virtualization*, the process of closing the gap between description of a thing and the thing itself. Some digital texts never existed in any non-digital form; others reached that condition after having been subjected to *digitization*, processed so as to become available for further processing—storage and retrieval, at least—by a digital computer. Additional processes to which digital texts can be subjected include: *localization*, adapting to the practices of group identity such as national language and currency; *personalization*, adapting to the preferences of an individual; *mobilization*, adapting to the limitations of a hand-held mobile device; *citation*, inserting a fragment of an older text into a newer text; and *isolation*, decapsulating a text's contents so they can be manipulated independently of their container.

A further large set of processes to which hypertexts are subject are those related to *collaboration*; in Chapter V, I identify and visualize fifteen sub-types of this process. Participants in *hyper*-collaborative processes can be artificial or natural intelligences; some digital texts, such as those created to aid the process of *transition* between one text and another, can contain within themselves the means of seeking human assistance. This built-in expectation of ongoing human support, I argue, is one of the important differences between paper and digital publication: for paper, the process of *publication* can be completed; for digital texts, especially those published on the Web, multiple levels of caretakers and controllers—webmasters, database administrators, forum moderators—must monitor and maintain the text throughout its existence. The reality of such monitoring and intervention by human caretakers is discounted in sources such as the *MLA Style Manual*, recommending caution in the use of electronic resources on the basis that most "are not refereed", unlike print publications "issued by reputable journal and book publishers that accept accountability for the quality and reliability of the works they distribute" (209). I argue that, while both print and digital publications are likely to be shaped by some level of supervision from their publisher, the refereeing process for a print journal is no more a guarantee of uniform quality control than is the mere existence of an online forum's moderator: there are *proactive* moderators, vetting every posting before allowing its publication; there are *reactive* moderators, interfering with forum activities only in case of complaint or obvious abuse; and there are *inactive* moderators, nominally responsible for the forum but actually taking no role in the publication process. For all texts, it remains the reader's task to judge the validity of the text and respond to it appropriately.

Transitional Forms: Digital Texts, Non-Digital Processes

Because some digital texts originated on paper, a digital text can sometimes be observed to refer to itself in ways that are appropriate for paper but not for digital media. Instructions related to the physical structure of a printed document, such as "Turn to page 53 for additional details" or "Use the back of this page to elaborate" are evidence that an electronic document was faithfully reproduced from an origin in print, just as instructions that can only be followed in a hypertext— "Click SUBMIT to process your selections"—can be evidence that a printed page is the wrong container, with textual contents not designed to be used on paper. Such mismatches can be errors, evidence that the results of an attempted adaptation to a new use were not carefully validated; they can also be intentional, accurately portraying a hybrid or transitional stage in the development of not only a text but the organization which publishes and uses it. Texts are human cultural products: inconsistency in the construction of a text can indicate disconnection within the culture that published it. Using textual structures as sources of insight into the developing social structures inhabited by the human creators of texts is the natural corollary of the projects described in the preceding chapters, exploring aspects of ongoing human involvement in textual development.

Figure 60 is an example of one such inconsistent text, a Petition for Extension of Time Limits published in Microsoft Word (.doc) format by Texas A&M University's Office of Graduate Studies at ogs.tamu.edu/forms/student-forms (Petition Documents).

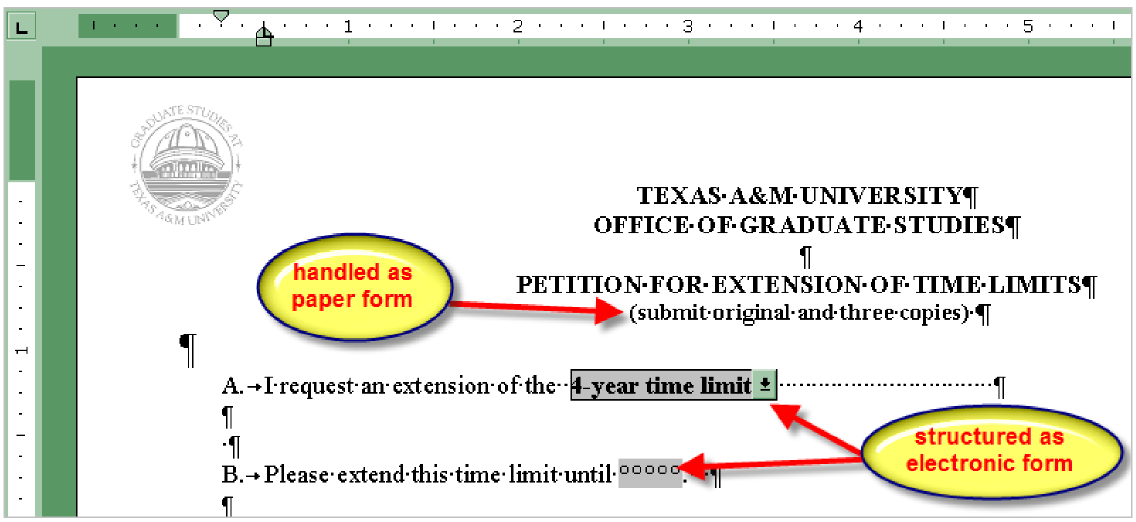


Figure 60: Instructions relate to human handling of a paper form created by this electronic form.

This text is not simply a digitized reproduction of an earlier paper form. Intelligent features have been added: most of the form is protected, with entry of personal data allowed only in designated locations; length limits are enforced for entered data, making it impossible to disrupt the form's one-page design; where the user can express a preference, an option must be selected from a pull-down menu rather than being entered as free-form text. However, although the form is made available via the Web and contains structural features taking advantage of its nature as a digital document, it contains instructions requiring it to be handled as a paper document: "submit original and three copies."

"Original" and "copy" are problematic concepts here, with different meanings for texts stored in paper or digital containers. For the user who obtains the form, completes it, and wishes to have it processed and approved, the digital text on the Web site is the "original," the source from which any copies must be derived. Multiple digital copies of the original digital document could be produced; their filenames and timestamps would differ from the original but their content would not, making the enterprise of creating copies a pointless one. Because the form provides locations in which to record manuscript signatures, the instruction to "submit original and three copies" must refer only to *paper* derivatives of the electronic form: the "original" becomes the one sheet of paper on which the form has been printed and manuscript signatures have been added, and the "copies" are images of the signed "original" paper, xerographically reproduced on other pieces of paper. The digital starting point on the Web becomes something pre-original; the older paper form from which the digital version was adapted can be imagined as an Ur-document, an imaginary pre-pre-original which may have existed for many years in multiple revisions during its lifetime on paper before it became the source for the digital document.

This text and others like it are participants in and evidence of large-scale transitional processes, in which a digital text must be transformed into a non-digital text, moving in a direction opposite to a previous digitization process, to enable further processing by non-digital means. Situations such as this arise because, quite naturally, organizations do not digitize holistically; paper forms can be replaced by digital forms much more easily than the associated human activities can be adapted to coordinate with that change. It is as if part of an office—the front lobby, with a display rack of helpful brochures, maps, and forms—is represented digitally but the remainder of the office is unchanged: to move past the front lobby with its display rack of forms and have one of the forms actually processed within the office requires that the form make the transition away from its digital "original" into a new, paper "original" that can be manipulated in non-digital ways. Paper provides a transitional medium between digital and non-digital document management tasks; the point at which users of a digital text are directed to paper—print, copy, fold, detach, staple, mail—is the point at which digital information connects to and depends upon non-digital processing. This is also a point at which artificial intelligences are excluded from further participation in a hyper-collaboration: non-digital paper is invisible to and unreachable by software.

Texts that are accessible to the full range of collaborations between human and artificial intelligences provide the broadest range of choices for textual manipulation and enhancement. For example, as introduced in the "What the Webmaster Saw" section of Chapter IV, computer-generated usage statistics can inform a human webmaster that an automated translation of a hypertext from its original into an alternative human language has been created on behalf of a human reader; the webmaster can produce the same automated translation, observe its low quality, and respond by initiating a high-quality (and high-cost) human-crafted translation of the hypertext to better accommodate future readers in that language; alternatively, the webmaster can take action to improve the quality of future automated translations by making structural changes such as extracting labels from images so that the labels become eligible for translation. Hyper-collaboration can also mean that human processes adapt to technological changes, creating new standards appropriate to new digital containers rather than preserving conventions developed to suit older paper containers. For example, the standard format of academic "papers" is an artifact indicating origin in print: wide margins and double spacing between lines creates "space" within the text for handwritten comments and markup by proofreaders and instructors, even though widely-used software tools for word processing, publishing, and teaching provide electronic means of incorporating such commentary without reserving physical locations on the page for it. Fixed "page" size is also an artifact indicating an intellectual origin and sometimes an intended destination in print: for digital texts, the unit of convenience is screen size, which varies per reader preference. A design goal for screen-intended texts is to minimize the reader's need to scroll, just as for print-intended procedural texts (cookbooks, repair manuals) minimal page-turning is a sign of careful design. As future publication technologies come into use, new expectations of appropriate textual design and formatting will arise, and new difficulties will complicate the processes of coordinating human and computer activities and the texts that represent and support them; I expect that, in the future as now, behavioral changes will lag behind technological changes and standards intended for older technologies will be preserved in texts created in new containers.

The Spike

Futurists speak of a point of singularity or a vertical asymptote, “The Spike,” beyond which it is impossible, using current knowledge and technology, to imagine what the future will be. Until then, the future is likely to be a continuation of present projects: computers become faster and cheaper and easier to use, more people have access to computers, more information is available through computers, computer networks become larger and faster and more pervasive. Beyond endlessly extending current trends, what is possible? In “Racing Toward the Spike,” Damien Broderick outlines some possibilities, including the rise of artificial intelligences as independent entities rather than human-controlled tools, people copying themselves into cyberspace rather than continuing to observe it externally, and the creation of “conjoined intelligence” (285) with groups of people collaborating by simultaneously coordinating and amplifying their thought processes. All other possibilities exist; by definition, it is not possible to guess what happens after The Spike.

In the nearer future, perhaps applications such as Google Earth (earth.google.com), which retain hypertext’s metaphor of travel but ask the user to freely fly over terrain rather than to choose among pre-defined hyperlinked paths through it, hint at logical developments; will “fly to” become a more common navigational instruction than “click on”? As described in *Quantum Computing* (Hervansalo) and elsewhere, work is in progress today toward a quantum computer, probabilistic rather than deterministic, designed to exploit all (at least thirty-two) quantum states of matter rather than the mere two states (positive or negative, on or off, in or out) currently in use;

with a computer architecture built to accept quantum conditions such as “simultaneously in and out” and “somewhat in and somewhat out,” and “sort of in, sometimes,” the current highly-containerized structure of hypertextual documents may make them stiff and unnatural, as limiting in their own way as printed documents are in theirs. Perhaps, as new systems are designed by people whose lifelong experience of texts of all kinds—newspapers, novels, banking records, street maps, love letters, grocery lists—is primarily digital, user interfaces will abandon multi-layered metaphors of paper-based practices (a page in a document in a folder in a cabinet; a chapter in a book on a shelf in a library) and offer currently-unimaginable methods by which people can interact with words and ideas.

I do not know what the next methods and systems of publication might be, nor how their internal natures may shape the human experiences of using and creating texts. I do expect that we will soon recognize a different kind of Digital Divide, understood now as the widening power gap between those with ready access to digital information sources and those without; there is also a related imagination gap, between those who use computers while mentally referring to non-computerized paradigms (the office desktop, the library, the jukebox) and those who use a computer as an independent technology with its own rules and methods, just as they drive an automobile without mental reference to driving a horse and buggy. The economically-created divide between those who do and do not have access to the Web may persist as long as access to information is treated as a luxury rather than a necessity; however, the population of readers who think of digital texts as substitutes for other, older texts will inevitably fade with time, making it possible to study hypertexts as objects with their own intrinsic natures, without prefatory discussion of how they differ from or resemble paper texts.

In The *Age of Spiritual Machines*, Ray Kurzweil posits seven stages of development applicable to any technology: the *precursor* stage, in which dreamers begin to imagine it; *invention*, in which a new technology is first demonstrated; *development*, in which it is protected and enhanced; *maturity*, in which it becomes accepted as part of a community; the stage of *pretenders*, in which alternative methods are proposed but fail to dislodge it; *obsolescence*, in which it gradually declines after having been superseded; *antiquity*, in which it becomes an artifact of past times rather than a currently-useful tool. Writing in 1999, Kurzweil places the printed book,

> in the stage of the pretenders, with the software-based "virtual" book as the pretender. Lacking the resolution, contrast, lack of flicker, and other visual qualities of paper and ink, the current generation of virtual book does not have the capability of displacing paper-based publications. Yet this victory of the paper-based book will be short-lived as future generations of computer displays succeed in providing a fully satisfactory alternative to paper (20).

I suggest that, for printing, the expectation of a "fully satisfactory alternative" is unrealistic and, even if it is never met, need not impede the continuing development of alternative publication technologies: for now, old and new publication technologies seem to be able to co-exist, serving different audiences by providing the same contents in different containers.

It may be some time before we hit The Spike and begin exchanging ideas by yet-unimaginable methods; for now, we have a great deal of work to do in understanding our current publication technologies, how they relate to each other, and how those relationships alter texts as we move them into containers for which they were not designed but into which we are determined to have them fit. By demonstrating some techniques of describing and criticizing hypertexts with the same rigor and granularity we apply to texts in older media such as print and film, I have suggested some possibilities in this direction; undoubtedly, the list of possibilities will grow as we continue to invent new forms of digital texts.

WORKS CITED

Aarseth, Espen J. “Nonlinearity and Literary Theory.” *Hyper/Text/Theory*. Ed. George P. Landow. Baltimore: Johns Hopkins UP, 1994. 51-86. Print.

Addey, Dave, James Ellis, Phil Suh, and David Thiemecke. *Content Management Systems*. Birmingham: glasshaus, 2002. Print.

Adobe. “Adobe – Photoshop CS3 Editions.” *Adobe*. Adobe. Web. 8 Aug. 2008.

Agualusa, José Eduardo. *The Book of Chameleons*. Trans. Daniel Hahn. New York: Simon & Schuster, 2008. Print.

“Barack Obama Keeps Fighting.” *New York Daily News*. 10 Jan. 2008. Web. 10 Jan. 2008.

Baskette, Floyd K and Jack Z. Sissors. *The Art of Editing*. New York: Macmillan, 1971. Print.

Bausch, Paul, Matthew Haughey, and Meg Hourihan. *We Blog: Publishing Online with Weblogs*. Indianapolis: Wiley, 2002. Print.

Berman, Jeffrey. *Narcissism and the Novel*. New York: NYU P, 1990. Web. 6 Aug. 2007.

Berners-Lee, Tim. “Information Management: A Proposal.” W3C World Wide Web Consortium. W3C. 1989, 1990. Web. 19 Aug. 2006.

Bolter, Jay David and Richard Grusin. *Remediation: Understanding New Media*. Cambridge: MIT P, 1999. Print.

Bordwell, David and Kristin Thompson. *Film Art: An Introduction*. 8th ed. New York: McGraw-Hill, 2008. Print.

Boxer, Sarah, ed. *Ultimate Blogs: Masterworks from the Wild Web*. New York: Random House, 2008. Print.

Broderick, Damien. "Racing Toward the Spike." *Prefiguring Cyberculture*. Eds. Darren Tofts, Annemarie Jonson, and Alessio Cavallaro. Boston: MIT P, 2004. 278-291. Print.

Buzetti, Dino and Jerome McGann. "Critical Editing in a Digital Horizon." *Electronic Textual Editing*. Eds. Lou Burnard, Katherine O'Brien O'Keeffe, and John Unsworth. New York: MLA, 2006. 53-73. Print.

CBS. "Smoke-Free Illinois Could Save Money and Lives." *CBS2 Chicago*. 28 Dec 2007. Web. 28 Dec. 2007.

Cisco Systems. "Open Systems Interconnection (OSI) Protocols." *Internetworking Technology Handbook*. Web. 19 Apr. 2009.

Committee on Scholarly Editions. "Guidelines for Editors of Scholarly Editions." *Modern Language Association*. MLA, 25 Sep 2007. Web. 5 Feb 2009.

Coover, Robert. "The End of Books." *New York Times* June 21, 1992. Web. 8 Dec. 2007.

Corel. "Paint Shop Pro Photo X2 – Photo Editing." *Corel Corporation*. Corel Corporation. Web. 8 Aug. 2008.

Coste, Rosemarie L. "Fighting Speech with Speech: David Duke, the Anti-Defamation League, Online Bookstores, and Hate Filters." *Proceedings of the 33rd Hawai'i International Conference on System Sciences – 2000*. IEEE. 2000. Web. 3 Mar. 2009.

Daragahi, Borzou and and Ramin Mostaghim. “Iranian protesters contest Mahmoud Ahmadinejad's reelection.” *Los Angeles Times*. 15 June 2009. Web. 15 June 2009.

Diehl, Stephan. *Software Visualization: Visualizing the Structure, Behaviour, and Evolution of Software*. New York: Springer Verlag, 2007. Print.

Diurno, Gianni. *Gidibao’s Café*. 20 Mar. 2009. Web. 22 Mar. 2009.

DotMobi. “Emulator.” *dotMobi*. Web. 5 Apr. 2009.

Eisenstein, Elizabeth. *The Printing Press as an Agent of Change: Communications and Cultural Transformations in Early Modern Europe*. New York: Cambridge UP, 1979. Print.

Elsom Research. “An Introduction to Nanosomes.” *Elsom Research Innovative Biotechnologies*. Elsom Research. 1997. Web. 21 Oct. 2007.

Folsom, Ed. “Database as Genre: The Epic Transformation of Archives.” Eds. Wai Chee Dimock and Bruce Robbins. *Remapping Genre*. Spec. issue of *PMLA*. 122.5 (2007): 1571-1579. Print.

Fussell, Paul. *Poetic Meter and Poetic Form*. Rev. ed. New York: McGraw-Hill, 1979.

Gandelman, Joe. “The Clinton’s Tone May Lose Them Civil Rights Veteran’s Endorsement.” *The Moderate Voice*. 11 Jan. 2008. Web. 11 Jan. 2008.

Gibaldi, Joseph. *MLA Style Manual and Guide to Scholarly Publication*. 2nd ed. New York: MLA, 1998. Print.

Gibson, William. *Neuromancer*. New York: Ace Books, 1984. Print.

Google. *Google Checkout*. 2 Apr. 2009. Web. Google. 2 Apr. 2009.

---. *Google News*. 29 Dec. 2007. Web. Google. 29 Dec. 2007.

---. *Google Translate Beta*. Google. Web. 1 Feb. 2008.

---. "News (Publishers) Help." *Google News*. Google. Web. 29 Dec. 2007.

Gore, Albert A., Jr. "Communications; Networking the Future; We Need a National 'Superhighway' for Computer Information." *Washington Post*. July 15, 1990. B3. Web. 11 Sep. 2008.

Hayles, N. Katherine. *Writing Machines*. Cambridge: MIT P, 2002. Print.

Haynes, Deborah J. "Bakhtin and the Visual Arts." *A Companion to Art Theory*. Eds. Paul Smith and Carolyn Wilde. Hoboken: Wiley-Blackwell, 2002. 292-302. Print.

"Helicopter Crash on A&M Campus Kills 1, Injures At Least 5." *Austin American Statesman.* 13 Jan. 2009. Web. 13 Jan. 2009.

Hervansalo, Mika. *Quantum Computing.* 2nd ed. New York: Springer, 2004. Web.

Hulse, Carl. "Civil Rights Tone Prompts Talk of an Endorsement." *New York Times*. 11 Jan. 2008. Web. 11 Jan. 2008.

Ink Corrosion Website. Royal Netherlands Academy of Arts and Sciences. Web. 23 Nov. 2007.

InkCor Project. "InkCor Home Page." *InkCor Stabilisation of Iron Gall Ink Containing Paper*. Web. 23 Nov. 2007.

IP-Lookup. *IP Address Lookup*. IP-Lookup. Web. 25 Mar. 2009.

"Judge Lynch's Work." *Chicago Daily Tribune*. Dec 31 1893. pg. 26. *Proquest Historical Newspapers*. Web. 3 Jul 2006.

Kahn, Ken. "Drawings on Napkins, Video-Game Animation, and Other Ways to Program Computers." *Communications of the ACM*. August 1996/Vol. 39, No. 8. 49-59. Web. 27 Feb. 2009.

Kalilich, Jordon. "Fake Cyrillic Generator." *The World of Stuff*. 29 Nov. 2008. Web. 4 Jan. 2009.

Kirschenbaum, Matthew G. *Mechanisms: New Media and the Forensic Imagination*. Cambridge: MIT P, 2008. Print.

Kiviat, Philip J. "Simulation, Technology, and the Decision Process." *ACM Transactions on Modeling and Computer Simulation (TOMACS)*. 1.2. ACM: April 1991. 89-98. Web. 3 Sep. 2008.

Kolko, Beth E., Lisa Nakamura, and Gilbert B. Rodman, eds. *Race in Cyberspace*. New York: Routledge, 2000. Print.

KompoZer. "KompoZer – about." *KompoZer*. Web. 16 Aug. 2008.

Koru, A. Günes, Dongsong Zhang, and Hongfang Liu. "Modeling the Effect of Size on Defect Proneness for Open-Source Software." *Third International Workshop on Predictor Models in Software Engineering (PROMISE'07)*. IEEE (2007). Web. 16 Feb. 2009.

Kurzweil, Ray. *The Age of Spiritual Machines: When Computers Exceed Human Intelligence*. New York: Viking, 1999. Print.

Landow, George P. *Hypertext: The Convergence of Contemporary Critical Theory and Technology*. Baltimore: Johns Hopkins UP, 1992. Print.

---. *Hypertext 2.0: The Convergence of Contemporary Critical Theory and Technology.* Baltimore: Johns Hopkins UP, 1997. Print.

---. *Hypertext 3.0: Critical Theory and New Media in an Era of Globalization.* Baltimore: Johns Hopkins UP, 2006. Print.

Langville, Amy N. and Carl D. Meyer. *Google's PageRank and Beyond: The Science of Search Engine Rankings*. Princeton: Princeton UP, 2006. Print.

Lévy, Pierre. *Cyberculture*. Trans. Robert Bononno. Minneapolis: U of Minnesota P, 2001. Print.

"Logoff." *New Equilibrium Skincare*. Elsom Research. Web. 20 Oct 2007.

McClain, John. "Coach, Team to Be Hurt Little by Punishment." *Houston Chronicle*. 14 Sep. 2007. Web. 14 Sep. 2007.

McGregor-Wood, Simon. "One Missile Too Many: Iran Doctored the Photo to Hide Failure of One Missle to Launch." *ABC News*. 10 July 2008. Web. 8 Aug. 2008.

Michelson, Bruce. *Printer's Devil: Mark Twain and the American Publishing Revolution*. Berkeley: U of California P, 2006. Print.

MobiForge. Web. 5 Apr. 2009.

Modern Language Association. "Frequently Asked Questions." *Modern Language Association*. 11 May 2007. Web.

---. *MLA Handbook for Writers of Research Papers*. 7th ed. New York: MLA, 2009. Print.

Moody, Glyn. *Rebel Code: Inside Linux and the Open Source Revolution*. Cambridge: Perseus Publishing, 2001. Print.

Nakamura, Lisa. "'Where Do You Want to Go Today?' Cybernetic Tourism, the Internet, and Transnationality." *Race in Cyberspace*. Eds. Beth E. Kolko, Lisa Nakamura, and Gilbert B. Rodman. New York: Routledge, 2000. Print.

"Namespace." *Wikipedia*. Web. 25 Mar. 2009.

Nelson, Theodor Holm. "Xanalogical Structure, Needed Now More Than Ever: Parallel Documents, Deep Links to Content, Deep Versioning, and Deep Re-Use." *ACM Computing Surveys*, 31.4es, December 1999. 1-32. Web. 17 Aug. 2007.

Neusner, Jacob. *The Mishnah. A New Translation*. New Haven: Yale UP, 1987. Print.

Ong, Walter J. *Orality and Literacy: The Technologizing of the Word*. London: Methuen, 1982. Print.

Opperman, Harry Earl III. *A Bibliography and Stemma Codicum for British Editions of Uncle Tom's Cabin, 1852-1853*. Diss. Kansas State University, 1971. Ann Arbor: UMI, 2001. Web. 8 Aug 2007.

Otoor. "Help Mw to Solve This Error Message from PayPal, This Recipient Does Not Accept Payments Denominated in USD." *OsCommerce Community Support Forums*. 10 Oct 2008. Web. 30 Nov. 2008.

Patrick, Tim. *Start-to-Finish Visual Basic 2005*. Boston: Pearson Education, 2007. Print.

Poetry Daily. Message to the author. 1 Sep. 2008. E-mail.

Preminger, Alex and T.V.F. Brogan, eds. *The New Princeton Encyclopedia of Poetry and Poetics*. New York: MJF Books, 1993.

Project Gutenberg. "Gutenberg: Scanning FAQ." *Project Gutenberg*. 13 Oct 2006. Web. 24 Oct 2007.

Rakitin, Steven R. *Software Verification and Validation for Practitioners and Managers*. 2nd ed. Boston: Artech House, 2001. Print.

Raskin, Jef. *The Humane Interface: New Directions for Designing Interactive Systems*. Reading: Addison-Wesley, 2000. Print.

Raymond, Eric S. "The Cathedral and the Bazaar." *The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary*. Sebastopol: O'Reilly Media, 2001. 19-63. Print.

"Record of 1893." *Chicago Daily Tribune* Dec 31 1893. pg. 20. *Proquest Historical Newspapers*. Web. 3 Jul 2006.

Ringlstetter, Christoph, Klaus U. Schulz, and Stoyan Mihov. "Orthographic Errors in Web Pages: Toward Cleaner Web Corpora." *Computational Linguistics* Sep2006, 32. 3. 295-340. Web. 15 Aug. 2007.

Rose [Rosemarie L. Coste]. "Expectations for Behavior in This Forum." *Topical Formulations Forum*. Journal of Topical Formulations. 26 June 2007. Web. 23 Mar. 2009.

---. "The Publisher Has No Control." *Hypertextual Ultrastructures*. Scribionics. 15 Dec 2008. Web. 21 Feb. 2009.

Rosello, Mirielle. "The Screener's Maps: Michel de Certeaus's 'Wandersmanner' and Paul Auster's Hypertextual Detective." *Hyper/Text/Theory*. Ed. George P. Landow. Baltimore: Johns Hopkins UP, 1994. 121-158. Print.

Ryan, Marie-Laure. *Narrative as Virtual Reality: Immersion and Interactivity in Literature and Electronic Media*. Baltimore: Johns Hopkins UP, 2001. Print.

Saffo, Paul. Preface. *24 Hours in Cyberspace*. Eds. Rick Smolan and Jennifer Erwit. Que Macmillan, 1996. 17. Print.

Sandvig, Christian. “The Structural Problems of the Internet for Cultural Policy.” *Critical Cyberculture Studies*. David Silver and Adrienne Massanari, eds. New York: New York UP, 2006. 173-190. Print.

Segal, Eliezar. “A Page of Talmud.” U of Calgary. Web. 31 Mar. 2009.

Shillingsburg, Peter L. *From Gutenberg to Google: Electronic Representations of Literary Texts*. New York: Cambridge UP, 2006. Print.

Steinbock, Daniel. *TagCrowd – Make Your Own Tag Cloud from Any Text*. Web.

Stijnman, Ad. “Iron-Gall Ink and Ink Corrosion.” *Archivum Lithuanicum (ArLi)* 4. 2002. 171-78. Web. 23 Nov. 2007.

Tarrant, R. J. “The State of the Art: Textual Scholarship in Classics Today.” *Scholarly Editing: A Guide to Researc*h. New York: MLA, 1995. 99-148. Print.

Texas A&M University. “AAPO First Friday Lecture – Dr. Ruben May.” *Division of Research and Graduate Studies*. Texas A&M U, College Station. Web. 23 Jan. 2008.

---. “Petition Documents – Graduate Studies.” *Office of Graduate Studies*. Texas A&M U, College Station. Web. 6 Feb. 2008.

---. “TAMU Department of English | Del Negro, Giovanna.” *Department of English*. Texas A&M U, College Station. Web. 22 Aug. 2007.

---. “TAMU Department of English | Dominguez Barajas, Elias.” *Department of English*. Texas A&M U, College Station. Web. 22 Aug. 2007.

---. “TAMU Department of English | O’Farrell, Mary Ann.” *Department of English*. Texas A&M U, College Station. Web. 22 Aug. 2007.

---. *Texas A&M University*. Texas A&M U, College Station. Web. 8 Jan. 2009.

United States. Cong. Senate. Subcommittee on Science, Technology, and Space of the Committee on Commerce, Science, and Transportation. *Hearings on the National High Performance Computer Technology Act of 1989*. 101st Cong. 1st sess. Washington, DC: GPO. 1989. Web. 15 Sep. 2008.

United States Department of the Treasury. *United States Internal Revenue Service*. Web. 24 Dec. 2007.

United States Patent and Trademark Office. “Home Page.” *United States Patent and Trademark Office*. Web. 7 Nov. 2007.

“User:ClueBot.” *Wikipedia*. 11 Oct. 2007. Web. 25 Mar. 2009.

“User:Ossmann.” *Wikipedia*. 9 Oct. 2008. Web. 25 Mar. 2009.

Van Susteren, Greta. “Gov. Blagojevich Goes 'On the Record': 'The Fix Is In'.” *FOX News*. Web. 27 Jan 2009.

Vilee, Claude A. *Biology*. 7th ed. Philadelphia: W. B. Saunders, 1977. Print.

VOA News. “Thousands Mourn Victims of Jerusalem School Shooting.” *Voice of America*. 7 Mar. 2008. Web. 7 Mar. 2008.

Weisman, Steven R. “Deal Is Offered for Chief’s Exit at World Bank.” *New York Times*. 8 May 2007. Web. 2 Sep. 2007.

“Welcome to the White House.” *The White House – President George W. Bush*. 16 Nov. 2007. Web. 16 Nov. 2007.

Wells, Ida B. *A Red Record: Tabulated Statistics and Alleged Causes of Lynchings in the United States, 1892-1893-1894*. Salem: Ayer Reprint, 1991. originally 1895. Print.

Wells, Stanley and Gary Taylor. *William Shakespeare: A Textual Companion*. New York: Norton, 1997. Print.

West, Thomas G. “Visual Thinkers and Nobel Prizes.” *Computer Graphics*. February 2001. 14-15. Web. 27 Feb. 2009.

Wikipedia. Web. 5 Apr. 2009.

Williams, Rick. “Theorizing Visual Intelligence: Practices, Developments, and Methodologies for Visual Communication.” Ed. Diane S. Hope. *Visual Communication: Perception, Rhetoric, and Technology*. Cresskill: Hampton Press, Rit Cary Graphic Arts Press, 2006. 31-56. Print.

Wimsatt, W. K. and Monroe C. Beardsley. “The Intentional Fallacy.” *Critical Theory Since Plato*. Rev. ed. Ed. Hazard Adams. Fort Worth: Harcourt Brace Jovanovich, 1992. 945-951. Print.

WordPress. *WordPress > Blog Tool and Publishing Platform*. WordPress.org. Web.

Žižek, Slavoj. “What Can Psychoanalysis Tell Us About Cyberspace?” *Psychoanalytic Review*, 91.6, December 2004. Web. 31 May 2008.

VITA

Name:	Rosemarie L. Coste
Address:	c/o Department of English Texas A&M University 4227 TAMU College Station, TX 77843-4227
Email Address:	mailto:rose@scribionics.com

Education:	B.S., Computing and Information Sciences, Trinity University, 1981 M.A., English, The University of Texas at San Antonio, 1998 Ph.D., English, Texas A&M University, 2009