

DIGITAL ACCESS TO SCHOLARSHIP AT HARVARD
DASH.HARVARD.EDU

HARVARD LIBRARY


The Origins of the Underline as Visual Representation of the Hyperlink on the Web: A Case Study in Skeuomorphism

Citation

Romano, John J. 2016. The Origins of the Underline as Visual Representation of the Hyperlink on the Web: A Case Study in Skeuomorphism. Master's thesis, Harvard Extension School.

Link

http://nrs.harvard.edu/urn-3:HUL.InstRepos:33797379

Terms of use

This article was downloaded from Harvard University's DASH repository, and is made available under the terms and conditions applicable to Other Posted Material (LAA), as set forth at
https://harvardwiki.atlassian.net/wiki/external/NGY5NDE4ZjgzNTc5NDQzMGIzZWZhMGFlOWI2M2EwYTg

Accessibility

https://accessibility.huit.harvard.edu/digital-accessibility-policy

Share Your Story

The Harvard community has made this article openly available.
Please share how this access benefits you. Submit a story

The Origins of the Underline as Visual Representation of the Hyperlink on the Web: A Case Study in Skeuomorphism

John J Romano

A Thesis in the Field of Visual Arts

for the Degree of Master of Liberal Arts in Extension Studies

Harvard University

November 2016

Abstract

This thesis investigates the process by which the underline came to be used as the default signifier of hyperlinks on the World Wide Web. Created in 1990 by Tim Berners-Lee, the web quickly became the most used hypertext system in the world, and most browsers default to indicating hyperlinks with an underline. To answer the question of why the underline was chosen over competing demarcation techniques, the thesis applies the methods of history of technology and sociology of technology. Before the invention of the web, the underline–also known as the vinculum–was used in many contexts in writing systems; collecting entities together to form a whole and ascribing additional meaning to the content. This early usage made the underline a natural choice, semantically, as a hyperlink signifier. The technological context in which the web was created also played a significant role in the use of the underline. Early computer systems were limited in their display capacities and the interfaces created on them were influenced by these constraints. Specifically, the NeXT computer on which Berners-Lee developed the web made underlining text very simple using its Interface Builder software. The combination of the existing semantic meaning of the underline in written language, coupled with the affordances of contemporary computer systems resulted in the commonplace use of the underline to signify hyperlinks.

Table of Contents

I. Introduction 1
II. The Histories of the Underline and the Visual Representation of Hyperlinks 9
The History of the Underline 9
Ancient History 10
Paper Systems 16
The Typewriter 19
Telegraphy 24
Visual Representations of Hyperlinks Before the Web 27
The Memex 30
The Typewriter, Part II 31
Early Hypertext Systems 33
Display Adapters 40
III. The Invention of the Web and Underline as Hyperlink Semaphore 50
Contemporary Hypertext Systems 50
HyperCard 50
Minitel 55
Gopher 57
NeXTSTEP 60
Tim Berners-Lee and the World Wide Web Standards 66
HTML 70
The Demonstration Browsers 76
Early Browsers 80
Mosaic and its Spawn 82
Usability and the Future 85
IV. Conclusion 89
Appendix 92
Bibliography 94

Chapter I

Introduction

The hyperlink is one of the most pervasive user interface elements on earth, featured on billions of web pages accessed daily by hundreds of millions of users. Despite its ubiquity, there has not been a complete investigation as to the origin and development of its visual representation on the web. In bringing to light the origin of the underline as a visual symbol for hyperlinks on the web, I researched the history of each of those domains: the underline, the hyperlink, and the web. I found three questions that were insufficiently addressed in the literature, one about each topic. The primary question–essentially, why did Tim Berners-Lee choose the underline to represent hyperlinks–will be addressed in Chapter III. Chapter II will address the other two questions, the answers to which should set the stage for the former and give insight into the milieu in which Berners-Lee was creating. The first of these questions relates to the underline: what did the underline represent before the invention of the web? To answer this question, I follow its use throughout the known history of writing and demonstrate that, while it has been used in many writing systems, it shares the common meaning of tying pieces of text together, allowing them to act as a group. This usage remains true across languages and various media of reproduction. Secondly, Chapter II will illuminate the visual representation of the hyperlink throughout the history of hypertext systems. Although eventually eclipsed in popularity by the web, early popular hypertext systems used a variety of modalities of indicating hyperlinks. An investigation of these systems

reveals that the use of the underline to represent links on the web was not the only possible choice. This investigation is also important because many other histories of hypertext choose to focus on the information architectural and information theoretical aspects of these hypertext systems, rather than the visual and presentational aspects. The answers to these two questions are interesting in their own right, but and they will also properly contextualize Berners-Lee's invention of the web and his choice of the underline.

The use of the underline on the web does not exist in a vacuum, but is part of an identifiable process within the discipline of graphic design. The history of graphic design spans many media and modes of mechanical reproduction–scribal transcription, movable type, offset printing, desktop publishing, web sites, and many more–each with their own distinct graphical identities and tropes. Each medium's aesthetic is shaped by the interplay between the technical limitations of the novel medium and innovation on the part of designers, but another important factor is the established tropes of extant media. Frequently, designers and inventors incorporate aesthetic traits from earlier media into their designs for new media, even if those traits originally arose for technical reasons that many no longer apply; this process is termed skeuomorphism. The term was first coined by H. Colley March in the 1889 edition of the *Transactions of the Lancashire and Cheshire Antiquarian Society* and refers to the ornamental use of material aspects of artifacts which became unnecessary due to technical improvements (166). Many such examples are documented in the history of print; this thesis examines how skeuomorphism continues to function in the age of web publishing. The "default" solutions that define skeuomorphism have practical application in that they ease adoption

by “priming” users for a particular choice. To this end, this paper demonstrates how earlier media influenced the aesthetics of early web design, specifically in regard to hyperlink, by charting the history of both the underline and hypertext systems up to the point of their convergence.

Early hypertext systems, and especially the web, created a new and distinct medium for sharing content and required novel graphic design treatments to convey their use. Hyperlinking is elemental to the medium and did not have direct precedence in previous media, and thus required an innovative design to separate it from surrounding text. Though the early web saw many different methods of indicating hyperlinks, a consensus coalesced around the underline as the default visual signifier. Hypermedia as a concept is generally traced back to Vannevar Bush’s *Memex*, a notional microfilm-based machine capable of creating and storing arbitrary linkages between pieces of information, akin to the workings of the brain (Bush). The terms describing such a system–“hyperlink” and “hypermedia”–were developed by Ted Nelson (Berners-Lee & Fischetti 219). Many offline hypertext systems were developed, but none flourished as a widespread, global medium until Tim Berners-Lee developed the World Wide Web (comprised of the HTTP protocol, HTML markup language, URI specification, and the WorldWideWeb demonstration browser) in 1990 (Berners-Lee & Fischetti 30). His development built on earlier methods of sharing content over the Internet, but made them visually accessible and freely linkable, all without the need for a central authority. Berners-Lee’s specification was intentionally minimal so as not to inhibit future innovation, and left questions of visual presentation open for browser developers to implement. As a result, early browsers such as UdiWWW used many different modalities to represent links on a

page (Obendorf & Weinreich 3), including the use of color, boxes, icons, and hotkeys. Berners-Lee's demonstration browser and its major successor, Mosaic, were strong influences on underline usage adoption, and by the time Netscape (1994) and Internet Explorer (1995) were released, the decision to use underlining was the de facto standard.

Many books have been written on hypermedia, but this study will focus on the visual aspects, which have been overlooked. For example, Jakob Nielsen wrote *Hypertext and Hypermedia* in 1990, presenting an overview of hypermedia history, but giving scant attention to presentational aspects of the hyperlinks of these systems. His 1995 follow up, *Multimedia and Hypertext* was released well after the web had taken off, but still glossed over the specifics of link presentation. The elision of this topic can be attributed to the amount of material covered, but I believe that insufficient attention was paid. Therefore, I intend to give a different type of history of hypertext, one centered on the display aspects of links, and demonstrating the material considerations which led to their creation.

The second chapter of this paper tracks the history of the underline in its usage in written language to give historical precedent to its eventual use on the web. In much the same way that Keith Houston does in his book *Shady Characters,* this chapter will explore the somewhat hidden history of the underline in all of its uses. This precedent extends back to ancient Rome, and beyond; the Romans used the *vinculum* (in the form of both under- and over-lining) to distinguish Roman numerals from letters in running text (spaces were not used to break up words). Later, especially into the Middle Ages, the vinculum was used to indicate numeral multiplication (Smith 60). Early manuscripts used underlining along with glosses and marginalia to highlight important passages in texts. Unlike other forms of highlighting, underlining is quite easy and can be performed on

paper with any writing instrument. The underline continues to be used for notation and emphasis by book readers and was adopted in areas related to hyperlink-like media, including in library science, where words were underlined to indicate the word by which a book was filed. The second chapter will show how the historical usage of the underline, coupled with the existing technological milieu, predisposed developers to select the underline.

The underline came into its own as a cultural force with the invention of the typewriter. Since typewriters generally contain only a single font, variants such as italics, bold (other than double-typing), small-caps, and other techniques were not readily available, so the underline was used to satisfy many needs. On a typewriter, creating an underline was just a matter of over-typing with an underscore, so it could be implemented easily without much modification to the machine. Monospaced typewriters made this very easy because each character was the same width as an underscore, but a more advanced technology was required to do this well on a later variable-width typewriter, such as the IBM Electric Executive Typewriter released in 1944 (“IBM Typewriter Milestones”). Perhaps not coincidentally, some of the first devices used as terminals to interact with mainframe computers were based on electric typewriters. The inherent limitations of using a typewriter-like device as the input/output device for a computer system shaped the development of early computer systems, and thus early hypertext systems.

As computers have replaced typewriters, and much printed matter as well, the underline has continued to be used, but the precise manner has evolved. Early on, web designers would sometimes use the underline to indicate emphasis (like they would on

earlier media) as well as for indicating links. The HTML specification allowed this with an underline tag. This caused confusion for users who quickly grew to expect underlined text to be clickable. Over time it has become best practice to avoid using underlining for purposes other than indicating links, showing how the ubiquity of hypertext has changed common usage ("Semantics, Structure, and APIs of HTML Documents"). Chapter III will discuss the evolution of the HTML standard, and how it shaped the visual representation of hyperlinks.

When Tim Berners-Lee developed the web on the then-relatively-new NeXT computer using the NeXTSTEP operating system in 1990, he chose to represent links with underlines, and thus set a precedent for other early browsers. Some other browser implementations, such as UdiWWW, chose alternative methods of showing links. Chapter III will track the influence that Berners-Lee's first browser had on the development of subsequent browsers, both through direct code transfer as well as stylistic influence. Some previous authors have generally attributed the ascendency of the underline (rather than just colored text) to the fact that many early computers–such as the NeXT–lacked color displays, although this argument greatly simplifies the many technological and cultural factors that led to the use of the underline (Obendorf & Weinreich 2). For instance, this argument does not exclude many other possible modes of representing a link that were not ultimately chosen, such as brackets, boxes, or icons, which were all used in some early hypertext systems. This thesis will demonstrate that the earlier use of underlining in other fields, such as library science, and in other media, such as the typewritten page, created an understanding of underlining that made its use to

represent the hyperlink natural and intuitive, leading to its eventual near-universal adoption.

This thesis will approach the topic through multiple theoretical lenses, owing to the diversity of topics covered. The research into the pre-web usage of the underline is primary historical, dipping into linguistics, mathematics, and the history and sociology of technology. When discussing the adoption of the web, a sociology of technology approach is used to explicate its success. Critically, I do not make an *a posteriori* argument about the success of the web: at each stage of its development I document its challenges, alternatives, and dead ends. Pinch and Bijker, in *Social Construction of Facts and Artefacts*, critique the history of science approach which they find overly focused on descriptive historiography, claiming that "preference for successful innovations seems to lead scholars to assume that the success of an artefact is an explanation of its subsequent development." As a result, they continue, some "[h]istorians of technology often seem content to rely on the manifest success of the artefact as evidence that no further explanatory work needs to be done." I will attempt to avoid this pitfall using the sociology of technology approach where "[t]he success of an artefact is precisely what needs to be explained. For a sociological theory of technology it should be the *explanandum*, not the *explanans*" (406).

The web exists dually as a technological invention and a graphical medium, so this study will also consider the aesthetics of the web, and the underline on the web, throughout. The underline bears information graphically and as Tufte defines it in *The Visual Display of Quantitative Information*, graphical excellence "consists of complex ideas communicated with clarity, precision, and efficiency" (51). While Tufte may seem

like an inappropriate touchstone when viewing the web as a medium–his work, after all, is on *quantitative* information)–the web is simultaneously a directed graph consisting of nodes and edges: a type of data structure. Tufte presents the concept of data-ink–the essential content of a graphical presentation–the loss of any of which would remove information. An efficient graphical representation of data will use a very high ratio of data-ink to total ink (93). I will show that the underline was leveraged to great effect by early browser developers to allow users to easily understand the vast complexity of the web with the use of a very simple signifier (high data-ink ratio), while also showing that competing proposals failed to improve on it.

I am undertaking this study to bring understanding to the use of skeuomorphism for good design–through helpful metaphors–as well as for bad design–in the case of lazy shorthands–as well as filling in an important lacuna in the history of the web. The exploration of the underline brings together sources from multiple areas of study–archeology, linguistics, mathematics–to chart how the underline has been used, and what it has meant throughout history. The underline is just one example of the many visual elements that we seamlessly interact with daily on the web, but to which we give little consideration. Considering the history of the underline also adds to web history by filling in the creation story of one of the most fundamental elements of web design, the hyperlink. This research provides insight about user interface creation that may be of help to those creating and improving user interface elements.

Chapter II

The Histories of the Underline and the Visual Representation of Hyperlinks

The History of the Underline

The underline and the closely related overline have a long precedent in the writing systems of the western world, as supported by archeological evidence from as early as 17,000 years ago. As the name implies, an underline is a horizontal line drawn below a piece of text, while an overline is one drawn over it. It seems that the usage of the line to connect a series of items to each other is a very natural affordance and has been used by many unconnected groups. Groups as diverse as the prehistoric populations of France, ancient Egyptians, Greeks, and Romans all used under- or overlines in their writing systems for various semaphoric purposes. The underline could owe its ubiquity to its simplicity: a line can be inscribed with a wide variety of tools including flint, reed, stylus, pencil, pen and ink, etc., and forms a combinative element of many writing systems. In fact, to be able to write Latin, Greek, Etruscan, Egyptian, or Sumerian, you must first be able to form a line; writing a letter such as “H” requires at a minimum two vertical lines and one horizontal line. While not every language system adopted the underline, any of them *could* have, so it is no surprise that is has been used as a component of many languages and with multiple different meanings. However, despite this variation in usage, there is a common thread: the underline is used to tie together pieces of writing. The underlining writer intends to signal to the reader that the text indicated is related in some way, thus overloading the text with additional layers of meaning, some of which will be

discussed below. The many uses of the underline in the millennia before the computer revolution have lodged it in our consciousness in a way that makes its use to indicate the hyperlink very natural.

Ancient History

In some contexts, the underline and overline are used interchangeably, and sometimes in conjunction: in Roman inscriptions, for instance. While one or the other is usually preferred in a given domain of usage, it is not uncommon to see them used interchangeably, as with the *vinculum* in mathematics. Within a given language or context where both the underline and overline are used, they do not tend to have distinct meanings. Presumably, this is due to the confusion this would cause in running text–is that an overline for this line of text or an underline for the previous? Therefore, in this section I will consider both types of lines as serving generally the same function, and may refer to them collectively as *vincula*, which speaks to their most common purpose.

The meaning of the word *vinculum* will help elucidate its history and meaning in the history of the use of the under/overline in writing. The word derives from Latin, and "comes from the family of words round the verb *sincere*, meaning 'to bind'. A *vinculum* is anything with which binding is done. Hence, a fetter, bond, chain or rope" (Birks and Descheemaeker 3). The term was also used in other contexts such as the Latin legal phrase "vinculum iuris" meaning "bond of law" referring to the important legal principle of obligations; a vinculum to a Roman was a critical and vital tie between entities, be they legal or textual. This definition describes very well the usage of the under- or

overline as an agglomerator, collecting entities together to form a whole, allowing it to be ascribed meaning beyond the content: a function that is a durable trait of the vinculum.

In mathematics, the vinculum has two main usages. First, it can refer to the line over or under a portion of a mathematical expression which indicates repeating decimals. For instance, five twelfths ($^{5}/_{12}$) can be represented in decimal notation as *0.41̲6* showing that the six digit repeats *ad infinitum*. This tool is used to achieve precise rendering of an expression while negating the need to use an approximation. The "bar" (which can contain multiple digits) serves two roles: first, it ties the numbers within together as a group, and second indicates that the group repeats forever. The vinculum here acts as an infinitely recursive function, taking a set of integers as an input and outputting a string endlessly repeating them. The other use of the vinculum is to group expressions in a manner similar to parentheses, affecting the order of operations of an equation. As in the recursive use, the bar, first and foremost, groups a set together.

The earliest documentation that I have found of the line being used in the capacity of the vinculum dates to an artifact from the Magdalenian era found in what is now Brassempouy, France. The artifact consists of an antler which was carved with a series of parallel lines. The lines are grouped into a series of divisions and separated by a perpendicular line (Ifrah 62). The perpendicular lines are thought to group the other marks together and represent an early use of tally marks. This usage presages "five-bar gate" system of marking off counts in groups of five: four verticals and a fifth horizontal to complete the grouping. This system allowed pre- or semi-numerate cultures to record quantities without the need for a robust system of number words or symbols (Ifrah 7). That the system is still in use today attests to its intuitiveness and usefulness, possibly due

to its human factors: five being the number of digits to a hand and also the outer bound of items able to be immediately quantified by sight without counting (Ifrah 7). The horizontal line functions as a vinculum by tying the first four digits (a hand) into an easily recognizable unit.

The vinculum was also used in Pharaonic Egypt and the Hittite Empire in their hieroglyphic writing systems. The hieroglyphic system used pictographic representations of objects and actions to document the world. Over time, shortcuts were taken to make for more concise and convenient writing, and representational images became more stylized. Plural words were indicated by a triplication of a given hieroglyph or by drawing three horizontal lines over a pictograph (Ifrah 94). Here, again, the lines act as function, multiplying the meaning of the symbols contained within. While the meaning is more literal and less symbolic than later uses of vincula, it is still a device apart from the language, even if it cannot be directly connected to later uses of the device.

The Greek language represents the solid beginning of the Western use of the vinculum, from which a straight line can be drawn to our current usage. The written number system of the first century BCE Greeks was acrophonic, representing numbers in writing by using the first letter of the number's name. For instance, the number five was "pente" (romanized) and was therefore represented by the Greek letter Pi (Π) (Ifrah 182). This system overloaded the alphabet with a second set of meaning, numerical in nature. While this overloading formed a convenient mnemonic device to aid comprehension, it did so at the expense of clarity in written documents. Because the same glyphs were used for numbers and letters (and since the language had not yet adopted the use of spacing between words) parsing a string of characters into the correct words was potentially

recondite. One could imagine the predisposition to “garden path sentences” this system must have caused. The Greeks eventually adopted a system of representing numbers sequentially in the order of the alphabet (A=1, B=2, Γ=3, etc.): if you knew your ABC’s, you could count. This system—which survives to present day in the form of Greek ordinals–had its advantages, but did not ameliorate the overloading issue (Ifrah 219). To solve that, the Greeks used the vinculum to signal to a reader that the under/overlined characters should be parsed numerically. This innovation can be found in extant inscriptions and documents, and greatly increases the readability of a text, without the need for a separate series of glyphs solely for numbers. Again, the vinculum ties the underset letters off as a group, imputing additional meaning to this group.

Other contemporary cultures developed an alternate method of representing numerals by using a separate glyph set. From a semiotic standpoint, the Greeks chose a system of in-band signalling: using the existing medium of their alphabet to express numbers transposed by the vinculum. Asian languages such as Japanese and Chinese used the out-of-band signaling approach by employing a totally different glyph set (medium) for their numbers[1] (Ifrah, Chapter 21) resulting in an implicit unambiguity of number writing. Despite alternatives, the Greek system formed the basis of Western numerical writing for centuries, perpetuating the need for continued use of the vinculum.

As in many areas of their culture, the Romans adopted the Greek system of overloading alphabetic characters with numerical meaning. In fact, they used the vinculum in at least two different manners, both numeric. First, the vinculum was

1. Approached from the perspective of McLuhanism, this raises the question of whether there is an appreciable difference in the reception of numbers in a language which uses the in- versus out-of-band signaling technique. Are the meanings of numbers shaped by their alphabetic pronunciation?

employed in the same manner as the Greeks to indicated numbers in running text from at least the first and second century BCE (Keppie 48). The Roman numeral system derived separately from the Greeks, and is thought to derive from the Etruscan number system. The numeral forms merged with Roman letters which shared visual similarity, resulting in the system that is now familiar to us through chapter headings and grandfather clocks– I,V,X,M, etc. (Ifrah 187). The Roman numerals were essentially homographs of their alphabetic counterparts. While the letters chosen differ from the Greek, the property of overloading is the same. In fact, the Romans would also use the vinculum when quoting Greek sources within an inscription, such as on the Library of Celsus while recording the career of Celsus (Keppie 75). This alphabetic parsimony had both disadvantages and advantages, as we will see at the beginning of the typewriter age when certain keys are used to represent multiple characters (e.g. lowercase "L" for the numeral one, and capital "O" for zero). The Romans were also creative in their use of the vinculum, not only offsetting whole integers, but abbreviating words where they could. Roman epigraphy reveals examples such as *duumvir* and *triumvir* (a political body of two and three leaders respectively) to be shortened to $\overline{II}VIR$ and $\overline{III}VIR$, the two (II) and three (III) standing in for the *trium-* and *duum-* prefixes (Ifrah, 198-199).

Roman use of the vinculum survives on many inscriptions on monuments, buildings, and documents, and influenced scholars of the middle ages in many ways, not the least of which is the orthography of inscribed Latin, which survives to the present as our serifed capital letters. Educated scribes and religious authors of the middle ages exulted Roman culture as a pinnacle to strive towards and adopted many of the quirks from the ancients. In Roman usage, the vinculum itself was overloaded with meaning.

Besides its use to set off numbers in running text, Romans used vincula in some contexts to indicate multiplication of integers. Most common was the usage as a thousand multiplyer: for instance, in the context of a mathematical tract, $\overline{VI}$ would equal six thousand with VI representing six and the overline multiplying it by one thousand. The bar also had other multiplicative uses. In the context of currency, the symbol for the denomination *denarius,* ~~X,~~ represented the currency unit was derived from its valuation of ten of its constituency currency, *asses* (Keppie 21). There is some debate over the prevalence of the vinculum as a thousand multiplicand due to that fact that much of the evidence of its use comes from transcriptions of documents made in the medieval period, which I will discuss next. Ifrah believes that the thousand multiplicand vinculum was in currency by the first century CE (198). He also cites as evidence from the early second century CE an inscription from Ephesus (in turn, cited from Cagnat 1899) that a vinculum-like character was used for myriad (10,000) multiplication. This character differed from a vinculum by having ends that turned down to a greater or lesser extent over the ends of the numeral (Ifrah 199). The multiplicative meaning was firmly in usage in the Middle Ages and found in mathematical tracts (Ifrah 194). Like the contemporary mathematical usage, these vincula and *vinculoids* act as functions on their inputs to mediate their meaning in a concise, albeit potentially confusing manner.

It is worth noting that the vinculum is not the only solution to disambiguating an overloaded numerical medium, and sharing common linguistic roots is not deterministic of how numerical signifying is achieved. The Hebrew language–which also derives from the same Phoenician roots as Latin (Ifrah 212)–also used alphabetic glyphs to represent numbers. But, rather than choosing an under- or overline, Hebrew uses a stroke at the top

corner of the last letter of a string of digits to indicate numbers in running text. Conversely, the Coptic number system derived from hybrid Greek-Egyptian roots and used letter-numbers for small values and vincula for numbers over one thousand (Ifrah 224). As stated earlier, pictographically-derived languages were more oriented towards the visual meaning of glyphs and tended to develop separate glyphs for numbers[2]. Despite these alternatives, the Greco-Roman system survived as the dominant numerical paradigm in the Western world until the adoption of Indo-Arabic numerals.

Paper Systems

The term information system likely brings to mind a computerized solution to organizing data, but information systems have existed for centuries. The first systems that would satisfy our modern definition used paper as their medium and the index card form factor, specifically, was instrumental in allowing for the efficient manipulation of data. The use of the index card to organize information is a relatively modern invention. Swiss naturalist Conrad Gessner proposed in 1548 a system of "slips of papers for bibliographic records" made "by writing down the entries on sheets of paper, underlining or writing in capitals the catchword, and cutting up the sheets into slips that could then either be pasted down in the required order" (Hopkins 384). While this early use of cards was intended as an interim process for creating a more permanent bound work, it shows the rudiments of the reference-based card system existed in the 16th century. Gessner's system was possibly the earliest example of hypertext-like information system using underlines to

2. Although these systems themselves tended to be overloaded with phonetic utility separate from their ideographic meaning, the initial sound of an existing pictogram being used to "spell out" a concept with no symbol of its own.

represent hyperlinks. A similar system was used specifically for library cataloguing by Gottfried Wilhelm Leibniz as Librarian of Wolfenbüttel between 1691 and 1699. He used slips of paper with title, author, and shelf location written on them to summarize the library's holdings, departing from the standard topic-based shelving system (Krajewski 22). The first true library catalogue system was the Josephinian catalog begun in Austria in 1780 as a result of an edict issued by Pope Clement XIV dissolving the Jesuit order. Its libraries were then expropriated by the Vienna Court Library which required a system to organize its newfound holdings (Krajewski 36–39). These innovations over centuries set the stage for the government card catalog produced by the post-revolutionary French government, using the playing card as a medium of recording books. The impetus for this catalog was the dissolution of the confiscation of their books by the state libraries (Krajewski 46). These library systems represented an early approach to what David Weinberger, in *Everything is Miscellaneous*, calls a second order organization–the items in the collection being the first order (17). By separating the thing itself from a reference to the thing, metadata was freed from the constraints of the physical realities of the bookshelf and knowledge could be organized in new and efficient ways. The card itself catalog itself is a type of hypertext, with cards acting (metaphorically) as hyperlinks to the books they represent, and also as links to other cards/books through the use of cross-references, the user of which flourished as card catalogs became standard practice in libraries across the world.

Melvil Dewey, motivated by a reformist spirit, is best known for his invention of the Dewey Decimal System, which sought to ease the organization of libraries by dividing the entire breadth of human knowledge into ten categories, and recursively

subdividing each of those categories into ten more (Wiegand 22). He aggressively advocated for his system through his various commercial enterprises and it was ultimately adopted by libraries throughout the country. Dewey was also was involved in the standardization of the dimensions of the card catalog and the distribution of related card-based technologies through the American Library Journal which he controlled (Wiegand 53). While best remembered for his library-related endeavors, his commercially oriented products were also widely adopted (Wright 41). In his influential book about library science practice, *Library School Rules*, Dewey prescribed a series of marks (which he called “checks”) to be made on catalog cards to highlight salient information (4). These checks consisted primarily of underlines and double underlines, but also included miniature brackets and were used to highlight the author or topic which the book is indexed by (22-32). The library community widely adopted this manner of usage of the underline.

The card catalog served both as an antecedent to and inspiration for digital hypermedia systems, as it shares many of its principles. The topic or author is underlined to indicate that the datum is salient and that relevant content exists elsewhere in the corpus. As Krajewski describes in *Paper Machines*, a book in which he draws parallels between library card-based systems and digital computers (going so far as to describe them as turing-complete), “cross-references are the crucial feature of a network” and the cross-references on library cards, frequently indicated by an underline, act as a hyperlink; just like on the web, "on the basis of keywords and short forms, every point of the index card box can refer precisely to another" (64). The use of the underline in this context both

continues its traditional usage and also forms a bridge to its future use in digital hypertext systems. This is especially true with the adoption of the typewriter to create these cards.

The Typewriter

The typewriter was the first practical invention that had the potential to replace the pen or pencil as the dominant means of writing letters. Many early users of the typewriter were amazed at its ability to replicate the look of printed material–the invention of which preceded the typewriter by over 350 years–while allowing the user to write faster than using script. While there were many false starts in its developments (many of which are described by Michael H. Adler in his book *The Writing Machine*), by the end of the 19th century, the typewriter had become a ubiquitous office feature. For primarily technical reasons, the prevalence of the typewriter resulted in the underline's use in far more situations than it ever had been previously.

Many inventors toiled to succeed at the common dream of constructing a device which would allow a user to create one-off documents resembling a printed page, the first patent for such a device being granted in 1714 to Henry Mill in England (Current 23). Like most new technologies, the list of progenitors to what we would consider the "final" product is long. This process, well explained in George Basalla's *The Evolution of Technology*, makes it difficult to fix the exact birth of what we consider to be the typewriter. Many early inventions could produce type on a page, but due to various mechanical limitations were impractical for widespread adoption. A common limitation was the relatively small number of characters allowed by that many early devices, constrained by the complexity of space and physical linkages. For example, the crude

Sholes typewriter of 1843 did not contain numerals, necessitating the spelling out of numbers or, more expediently, utilizing Roman numerals (Current 25). This centuries-old numeral system became useful again for exactly the reason that the vinculum was previously needed: the overloading of a limited character set with additional meaning. Whereas it was a potential disadvantage in Roman writing, it was used expeditiously in this new mechanical circumstance.

Despite the many claimants and a long history of similar inventions, Richard N. Current in his book, *The Typewriter*, makes a convincing case that Christopher Latham Sholes's invention of 1872 was the first device to contain all of the critical elements of the modern typewriter, allowing it to become commercially successfully. These innovations include the auto-advancing platen, end-of-line bell, and QWERTY keyboard layout (Current, 50-55). Like modern typewriters, this device used a series of keys to effect the impression of inked metal onto a sheet of paper, thus approximating the effect of movable-type printing, the dominant mode of textual reproduction in the western world since the fifteenth century. However, the typewriter varies from printing in at least two critical ways. First, the typewriter (until innovations by IBM in the mid-twentieth century) used monospaced fonts, meaning that each letter, number or symbol was spaced at an even width. Secondly, each typewriter was physically limited in the number of characters available to type. Each of these differences has an important influence in the development of the underline, and a direct relation to the early computer age. This limitations of a small character set continued on many early computer systems, although for different technological reasons.

Monospace type is more convenient to engineer on a typewriter because the escapement which advances the platen is only required to move a fixed distance for each key press.[3] Whereas letterpress systems allow type of any width and variable kerning through physical means, typewriters required even spacing with characters often rendered in ways which filled up more space (e.g. "`i`" rather than "i"). This monospace system negated the traditional advantage of the italic in letterpress printing, which is the ability to set text more closely without reducing legibility by lowering point size, an innovation generally attributed (for instance, by the *Encyclopædia Britannica*) to Aldus Manutius in his printing of Virgil around 1500 ("italic"). Therefore, the typewriter removed one of the common modes of setting off sections of running text in print. Boldface, or blackletter type, was also absent in the limited character sets of early typewriters, although a facsimile could be produced by repeatedly overtyping a letter. The underline, however, was not impeded by this monospace limitation.

An even more important difference from typesetting was the limited number of characters available on a given typewriter. The movable-type printer was only limited by the number pieces of type owned by the printer–and even then, woodcuts could be produced to fill in any gaps in metal type. The relationship between the available character set and mechanical complexity of early typewriters was somewhat linear: more characters yielded a proportionally greater number of moving parts. With each character inclusion precious, much thought was given to which characters were omitted, and which were critical or could be overloaded with meaning. Sholes' 1872 typewriter had only

3. Many typewriters of this era used a spring-loaded carriage, holding the type, which travelled across the the page letter-space by letter-space by means of a gear called an escapement. The piece of paper was held still on a support called a platen.

numbers, a few punctuations marks, and letters which lacked the ability to shift between the cases–which would be invented in 1878. This limitation inspired Sholes' business partner, James Densmore to conceive of a special underline character—or combination bracket-underline—to indicate that the character was to be interpreted as uppercase (Current 38). This usage harkens back to the Roman *vinculum* in that it overloaded a character set with additional meaning, and is an early example of the underline as typewriter feature, even conceptually. This device also eschewed separate keys for the numerals "0" and "1," as they could be plausibly rendered by the "O" and "l" (lowercase "L") keys, respectively (Wershler-Henry 154). These examples illustrate how the mechanical limitations of the machine shaped the visual design of the documents they produced, a theme that will recur in the nascent stages of the development of hypertext.

The underscore key on the typewriter was a fixture of the keyboard by 1881, documented as a character on the "Caligraph" typewriter developed by George Washington Newton Yost (Adler 176). Underlining was achieved by backing the carriage up one space, then typing over the intended character with an underscore. The underscore was set below the baseline of the regular type and would thus form an underline. Because each character was monospaced, the underscores were connected, forming a continuous line. This scheme was to be used on subsequent typewriters and, as we will see, shaped telegraphic standards.

These two limitations explain why the underline quickly became an important feature on typewriters whereas it was not on the printing press: underlining was the only available form of accentuation of text available on the typewriter. A printer had multiple tools at his disposal to set off text; he could use a larger font size, *italic type*, **stronger**

type, blackletter type such as fraktur, SMALL CAPS, or any manner of ornament or wingding–the *manicule*, "☞", was frequently used for this purpose (Houston 172). Since each character was chosen from a typecase and inserted individually, no continuity between characters in a set was required. These variants were used frequently, and with many different meanings varying according to geography, subject, and time period. The other dominant mode of written communication, handwriting, also has its many modes of expressiveness and is even less constrained than the printing press with its ability to shape each letterform differently, switch between hands, or add extraliterary annotations such as curlicues, doodles, or marginalia.

None of these tools were available on early typewriters, but the need to produce an expressive document had not changed. To fill this role, the underline was a convenient surrogate. Adding but one more key allowed any other character on the machine to be modified, in a way doubling the combinative glyphs available. In practice though, the underline is more useful when deployed on the word- or sentence-level. The underline was used in many capacities, encapsulating the use cases of the previously described letterpress variants. The underline was used to give emphasis to a word or phrase by drawing the eye to it, either because of the additional visual of the printing, the Gestalt law of grouping, or the dissimilarity from the text around it. This principle is also used to great effect on the web, where best-practice advice, backed up by eye tracking research by Nielsen, et al. shows that the eyes saccade between underlined links on a page[4] (*Eyetracking Web Usability* 144).

4. This raises the question of whether we are primed to look for underlined text due to its conventional use to demarcate important content or an innate psychophysical predilection for the underline. A possible, although likely difficult to execute experiment, would be to test the ability of an underlined

The typewriter also fostered the use of the underline to represent what would traditionally be represented with italics, for instance titles of books. *The Chicago Manual of Style*, which was first printed in 1906, has been consistent in its use of italics for book titles, however, by the 1982 (13th) edition, in recognition of the common use of typewriters it issued advice on best practices of using the underline to indicate emphasis (*Chicago Manual of Style*, various editions). So while formal contexts that called for typesetting still preferred other forms of emphasis, the underline flourished as a signifier in the typewriter-dominated world. In this regard, the underlines function as a link or reference to the book in question, a very hypermedia-like approach. If the world of books is envisioned as a network domain, the books are nodes and references between them function as edges forming a web-like structure. The use of the underline representing titles presages the web's hyperlinks. The specific layout of the typewriter keyboard proved instrumental in shaping the telegraphic and eventual computational future of character sets.

Telegraphy

Like the typewriter, the telegraph has a long and incremental history, all of which is described amply elsewhere. Over time, it reshaped the world's communications paradigm, allowing near instantaneous communication over hitherto unprecedented distances. For the telegraphic *incunabula*, communications companies desired a way to automate and simplify the operations of the telegraph and remove the need for an operator to cipher and decipher the code (generally Morse code), which formed an

---

phrase to attract visual attention among a population which has not traditionally used underlining it its writing system.

expense as well as a bottleneck as transmission speeds increased. There was an overlap between typewriter development and this quest: Christopher Latham Sholes showed his nascent typewriter to inventor Thomas Edison in 1870 (Wershler-Henry 69), but Edison was unimpressed with this crude machine, convinced a device of his own creation would be superior. Ultimately, Edison did not create a successful typewriter, but instead his invention became the stock ticker (Adler 32). Other inventors did have more success merging the typewriter telegraph and and allowing telegraphic code to be directly converted into the written word. The first widespread standard for this was patented by Frenchman Jean-Maurice-Émile Baudot in 1874; his system used a series of five binary digits–allowing up to 32 combinations–to represent the alphabet in a machine-decipherable format. A shift code allowed this set to be be expanded with another 32 symbols such as numerals and punctuation (Fischer 1). This original character set had no facility to represent formatting of any kind, as its purpose was just to convey terse messages. This sufficed until the dawn of the computer era when, just as happened with the typewriter, the desire for a more expressive character set put pressure on standards-makers to expand the available symbols.

As telegraphic hardware began to be used to interact with automated and eventually computerized machinery there was a push to expand the telegraphic character set. No longer was a human the only ingester of telegraphically transmitted data. For example, the Murray Printing Telegraph patented by Donald Murray in 1899 used a five-bit code similar to Baudot's to print Roman alphabetic characters from a punched tape. His code allowed extra shift codes to further expand the number of expressible characters (Fischer 3). The various competing code standards posed a problem for international

communications organizations; without a standard code, devices could not communicate intelligibly. The differing alphabetic requirements of the various languages exacerbated this Tower of Babel scenario. The Comité Consultatif International des Communications Télégraphiques (CCIT) met in 1926 to hash out a global standard telegraphic character set (Fischer 6). The resultant International Telegraph Alphabet No. 1 and successful Alphabet No. 2 remained in use until the mid 1950s, at and which point digital computing required a more advanced character set. In 1963, ASCII code was approved, expanding on an early internal code used by IBM; ASCII contained letters, numerals, symbols, and control characters, but was still very basic and akin to early telegraphic code. However, this code was the basis for the first code that included the underscore character (Fischer 21).

The successor to the CCIT, the International Telegraph and Telephone Consultative Committee (CCITT), used ASCII as the basis for its own proposed standard, which was more internationally focused than ASCII. Included in the CCITT standard is the underscore character as well as combining diacritics: characters that would be added to other characters to change their meaning (Fischer 24). For example "A̲" would be sent as as string of three signals "A", *backspace*,"_", which would print an "A," back the writehead up one space, then overtype with an underscore, then advance once again. This same principle was used for accented characters, e.g. "é" would be sent as "e," *backspace*, "´". Other combining characters included tilde (˜) and circumflex (^). This system allowed for a reduced number of characters in the set without limiting the ability to produce variants. These changes ultimately made their way back into ASCII in 1967 and were heavily used in the development of computer systems (Fischer 28). The use of

“combining diacritical marks” presages the invention of the Unicode character standard decades later which vastly expanded the characters available to computer software allowing for the incorporation of non-Roman alphabets and characters sets, as well as special purpose symbols. Unicode uses the more efficient method of storing diacritics as separate character entities; this is especially important in languages which make heavy use to diacritics in constructing their letters (“The Unicode Standard Version 8.0 – Core Specification."). Unicode, which is now used on all modern operating systems, also includes a combining underline which functions in the same manner as the diacritic.

This section has demonstrated that the underline has existed through the majority of written human history, from early clay tablets to tablet computers. Its usage has varied in the specifics, but has always been a tool to tie content together and give it extra meaning. The next section will discuss the various way that links have been represented in hypermedia, including but not limited to the underlining, which will give insight into its use on the web.

Visual Representations of Hyperlinks Before the Web

As the first part of this chapter demonstrated, the desire and need to crosslink knowledge was felt long before the invention of the modern computer, especially as exacerbated by the flood of books produced after the commercialization of the moveable-type printing press. Publishers attempted to aid readers in navigating this sea of knowledge with the development of reference works with organizational tools such as outlines and indices between the 16th and 18th centuries (A. Wright 23). The underline

was used in many different contexts, but was frequently used to show a linkage. Many methods of constructing linkages were concocted and many means of indicating those linkages were tested. Card catalogues attempted to order the world of knowledge and used underlined index words to indicate to a user that more information existed on a topic and created a web-like structure. However, these systems were limited by human factors such as retrieval speed or physical space requirements. Also, systems tended to index only like content (e.g. books in a library, or a business inventory). While these systems were important innovations, the pace of knowledge creation did not wane, and new systems for cross referencing a broader range of knowledge were increasingly in demand.

This section will discuss the history of hypertext leading up to the invention of the web, specifically focused on the visual representation of the hyperlink in these systems. Next, it will show that many methods of highlighting links were used and that the underline was not the only or obvious choice for Tim Berner-Lee based on extant systems, and that the factors leading to the use of hyperlink signifiers were strongly influenced by the affordances of the computer environments in which the systems were developed. The determinism of hardware reflects the findings of the first part of this chapter which showed that many systems used the underline for reasons that were more technical than aesthetic or philosophical. This chapter ends with the invention of the web because it very quickly became the dominant global hypertext system.

Choosing a place to begin is non-trivial: there are many contenders to the first hypertext system. As George Basalla demonstrates in *The Evolution of Technology* by drawing a metaphor between Darwinian evolution in living organisms and the progression of artifacts that result in our current technologies, upon close inspection,

technological change generally happens not through well documented “inventions,” but incrementally and without fanfare over the course of time. The well known inflection points of history look much smoother under a microscope. But a choice must be made. I will begin with Vannevar Bush’s Memex, but there are other claimants to the title of first hypertext system. Alex Wright argues in his book, *Cataloging the World: Paul Otlet and the Birth of the Information Age*, that the eponymous Belgian librarian was the progenitor of hypertext systems. He cites for evidence Otlet’s 1934 proposal for a system of “electronic telescopes” to allow “people to search through millions of interlinked documents, images, and audio and video files...” using “desktop workstations — each equipped with a viewing screen and multiple movable surfaces— connected to a central repository which would provide access to a wide range of resources” which he called "réseau mondial” or global network (Wright 8). Adding weight to Wright’s argument for Otlet over Bush is the observation that Otlet’s 1934 proposal was predicated on networked systems whereas Bush envisioned solitary workstations (Wright 257). However, Otlet’s system was theoretical: he did not propose the specifics of its implementations and also his work was not widely disseminated due to the Nazi invasion of Belgium in 1940 and the destruction of confiscation of much of the Belgian library system (Wright 12). Perhaps most relevant to this study, Otlet was not explicit in the details of the user-interface of his system. Wright also puts forth Emanuel Goldberg, the inventor of many microfilm technologies, as a possible contender, and in fact Goldberg’s patent for an optical retrieval system based on microfilm technology was seen by Vannevar Bush and its prior art prevented Bush from patenting his Memex concept (Wright 208). However, Goldberg did not widely publish his vision or make a great

impact with his device, so his is more of an interesting footnote. As a result of the very wide publication of his concept and the explicit explanation of its user interface, Bush's Memex is the best starting point for a discussion of the visual representation of the hyperlink in the digital era.

The Memex

With the growth up publication unabated in the years in the years leading up to the Second Word War, the pressing need to develop new organizational systems led to the seminal publication of the essay "As We May Think" in 1945 by Vannevar Bush, American engineer and military research and development administrator. He made the observation that it was becoming impossible for scientists to keep themselves aware of all new research being published and to organize their notes and resources (Bush). Coming from a background in computer science, Bush proposed a hypothetical system called the "Memex" which would allow media of all types to be ingested, stored on microfilm, and easily retrieved on a screen by the user. Critically, the system would allow "associative indexing" using links or "trails" to be defined between any pieces of content by the user, in a manner similar to how our own brains connect memories. Such a system, he asserted, could increase the efficiency of the user to organize and retrieve his thoughts about and connections between content. This system is cited as the first description of what would become known as "hypertext" (Bardini 39).

In his essay, Bush describes how the Memex could function by extrapolating from contemporary technological developments, especially the then-astonishing storage capacity of microfilm. The navigation of user-defined links would have been effected

through the use of buttons and levers under a screen on a desk, presaging later computer terminals, if not the visual implementation of later hypermedia systems. The links would have been stored in a machine-readable format on the microfilm and presented as a list to the user on a per-page basis, instead of a per-word or per-arbitrary-section basis as in later systems. Despite differences in the visual representation of hyperlinks, the germinal concepts of the web were contained in Bush's essay. While it would be decades before the computer technology existed to attempt to implement a system as complex as the Memex, computer research accelerated with the invention of the integrated circuit, allowing hitherto inconceivable complexity to be achieved. Bush's writings were especially prescient as he wrote before transistors were even commercially available and before Gordon Moore described the exponential growth pattern of processing power (Hiltzik 89). The importance of the Memex was not technological–it was based on existing technologies, and then only discussed in generalities–but in bringing the idea of hypertext to a wide audience. It would be for later inventors to figure out how to make it work, and how to address the presentational challenges raised by the conceptual underpinnings of hypertext.

The Typewriter, Part II

Bush's grandiose vision of the Memex and concrete realities of hardware took many years to bridge but in the meantime, technology did advance, with exponential growth in transistorized computer processing and memory, perhaps outstripping innovation in interface technology. Existing technology was adapted to the needs of innovation, and, of particular relevance to this study, was the incorporation of the

typewriter as a computer peripheral to fill the needs of input and output (I/O) on rapidly improving computers. Developed as business machines, many computers in the 1950s and 1960s used a modified typewriter for I/O, replacing earlier systems that used punch cards and tape. IBM's very popular typewriter models 2740 and 2741, released in 1965, were based on the IBM electric typewriter with additional computer controlled actuators. The sensors, actuators and electronics added to the machine allowed the devices to function either as typewriters or as "computer-controlled interactive terminals… in time-sharing applications" (Pugh 581). This meant that multiple users would submit commands via their terminals to a remote mainframe computer, which shared its processing power among all of them, frequently resulting in slow response times. In a sense, the time-sharing configuration was a cross between telegraph and a computer, and as previously mentioned, the interplay between the typewriter, telegraph, and computer influenced the character code standard, ASCII.

During the development of the ASCII standard computer, programmers weighed in on character inclusions that would benefit and ease the use of their language of choice and its peculiarities. For instance, programmers of the ALGOL language pushed for inclusion of arrow symbols in the new standard because that was traditionally how certain functions were represented in that language (Fischer 14). The IBM printing terminal, based on the Selectric line, was especially well suited to printing alternative character sets because of its interchangeable type ball (Pugh 366). Also known as a "golf ball," the type for the machine was molded onto the periphery of a sphere connected to the keys by a complex series of linkages. Unlike traditional typewriters with an arm for each pair of letters, the typeball system allowed the type set to be easily changed by

swapping out the ball itself (Nicolson 1). Despite some flexibility with switching a typeball, these machines were still limited to a finite number of available characters (88 or 96 in IBM's case) and no formatting to speak of, other than the use of the backspace-underscore form of underlining (GP IBM Selectric Typeball Catalog).

The typewriter was drafted to serve as the input-option layer of early computers because it was readily-available, easily interfaced with and adapted to the needs of digital computers, and comfortable and familiar to office users. However, the influence between the machines was bidirectional: the physical constraints of the typewriter shaped the types of hardware built and programs written by engineers. And while the typewriter was not an optimal interface device (for instance, forcing computer programs to keep track of shift cases and line lengths) it was a convenient device and allowed computer developers to more quickly innovate by using the computational hardware. This pattern repeats again and again, including in the development of the web: aesthetic and user-interface factors are often subordinated to the inclination to use hardware defaults.

Early Hypertext Systems

Vannevar Bush's 1945 essay described both a piece of hardware and vision of a functionally-infinite stockpile of documents which could be arbitrarily linked to one another. Bush termed the hardware the Memex, but the organizational concept lacked a name until 1962 when Ted Nelson coined *hypertext* (Bardini 39). Hypertext, according to Nelson, referred to non-linear text which could be freely navigated by a user, and formed the basis of a system Nelson envisioned called Xanadu, similar in many ways to the Memex (Barnet 4). Though Xanadu was not built, Nelson was involved in one of the

seminal projects in early hypertext, the Hypertext Editing System (HES). The term hypertext, and the related hyperlink and hypermedia, are foundational to the study of all manners of related systems which can store and navigate information. The terms unify systems based on disparate technology (microfilm, digital computing, even–as Markus Krajewski argues–paper) into one field of study, and will provide a lens to investigate many different systems up to and including the web. What follows is a summary of the most important hypertext systems and their salient innovations, both hardware and software, which shaped the course of hypertext.

Before the mid-1960s, commercial computers, having grown out of business needs, were large, cumbersome, and used to execute programs as batches (Hiltzik, xxi). By the mid-1960s, many computers still used batch processing–in which a program was loaded and the computer allowed to work on the given problem until completion–but interactive or "online" interaction was beginning to be used due to geometric increases in the affordability of computing power and memory, as later described by Moore's Law in 1965 (Barnet 3). The IBM 360 system was a dominant player in the time-sharing market, and despite not being the spiritual successor of Vannevar Bush's Memex, was important in the development of early hypertext systems. It could be equipped with a light pen–a device which allowed the user to interact directly with the output display by using an optic sensor to detect when the electron gun crossed the indicated spot on the cathode ray tube. This interface gave the IBM 360 what Pugh described as the first "operator-display interaction" (599). For the first time in a commercial system, a user could point directly at the screen and effect an action on the indicated object, which is a very powerful metaphor as most interactions in the world work this way. Most importantly to the history

of hypertext, an IBM 360 system was used for the development of HES which also used a light pen for interaction. Developed at Brown University around 1967 by a team led by Andries Van Dam and involving Ted Nelson, “HES was the world's first word processor to run on commercial equipment. It was also the first hypertext system that beginners could use, and pioneered many modern hypertext concepts for personal use” (Barnet 3). Making a physical connection with a target object is a much more intuitive metaphor than typing an abstract command into a machine, relying on a frequently-arcane grammar. The direct interaction technique of the hyperlink on the web (albeit through the mediation of the mouse) can be seen as built on this type of metaphor.

Bush’s “As We May Think” also had a great impact on the work of Douglas Engelbart, who cited Bush’s essay as his motivation to begin work on his revolutionary, Memex-like computer system (Hiltzik 62). Engelbart joined the Stanford Research Institute (SRI) in 1957 to work on research computing projects. Stanford University of Menlo Park, California spun off SRI in 1947 as a private, non-profit organization. Its mission was to perform research and development projects in conjunction with business, academic, and governmental bodies, and while most of its staff’s time was spent working on contracted research projects, Engelbart was primarily interested in developing a novel interactive computer system (Bardini 14). Created in conjunction with Bill English, Dave Hopper and Robert Bates, who all joined SRI around 1964, the new system was to be known as NLS or “oN-Line System” (Bardini 120). The NLS was funded by ARPA-IPTO, the federally-funded research arm of the Advanced Projects Research Agency for Information Processing Techniques Office led by J. C. R. Licklider for whom Engelbart's vision of creating a system to augment human intelligence resonated (Bardini 23).

Engelbart envisioned the NLS as being an interactive system based on commercial hardware. While previous “personal computers” existed with an interactive, real-time interface, they allowed all of the machine’s computer power to sit fallow until the user issued a command. Other large computers were used for batch operations where programs were run to completion sequentially. Instead, the NLS used a time-sharing configuration to allow many users to interact with the computer simultaneously, and with a high degree of responsiveness, which was made possible by an operating system and hardware which would allow users to share the computer's processing resources and memory using a technology called “memory paging” (Bardini 125). The innovations in memory paging came from Scientific Data Systems (SDS), a company which also had involvement in early hypertext systems.

SDS’s 940 computer, which ran an operating system developed by Project GENIE at UC Berkeley, was chosen for Engelbart's interactive computer project because it supported time-sharing and memory paging, facilitating the interactive environment Engelbart envisioned (Bardini 123). The NLS software on the SDS 940 had two modes of interaction: teletype and “special delivery channel.” The teletype (in practice the Teletype model 33 and 37 were used) provided a much more limited level of interaction with the system. According to Bardini:

> The teletype terminal, a distant offspring of research in type printing for telegraphy, had been the standard interface for time-sharing systems since the early 1960s. It was basically a typewriter transformed for telegraphy input and printing. . . (127)

However, the special delivery channel mode based on CRT technology was a revolutionary new method in human computer interaction and allowed the interaction techniques that have become pervasive today. In fact, in the forty years between its

demonstration and the rise of touch based-computers such as the iPhone, the mouse-keyboard-screen combination was synonymous with personal computing.

Existing CRT technology used point-plotting or vector systems to convert instructions from a computer to an image on the display. These systems were limited in their ability to display graphics or complex imagery. However, more advanced systems incorporating a frame buffer, which, while producing much better imagery, required a great deal of memory, resulting in prohibitive costs (Bardini 131). To solve this problem, Engelbart implemented a hybrid system, which used a CRT mated to a television camera to convert the vector images to a scan-line signal compatible with normal television monitors (Bardini 133). The CRTs were small, the hardware to run them was complex and based on vacuum tube technology, and their output abilities were limited (Bardini 132). Despite these limitations, the NLS allowed the combined use of imagery and text at a reasonable cost and enabled SRI to present it in the first-of-its-kind video conferenced, real-time presentation to the ACM/IEEE-Computer Society Fall Joint Computer Conference in San Francisco, later dubbed the “Mother of all Demos,” in 1968 (Bardini 138). SRI’s demonstration marked the first public display of the computer mouse and a functional hypertext system. At the time, each one of these innovations could have impressed a crowd, but demonstrating them all together was revolutionary. The long influence of this presentation can be seen in the current paradigm: every time one uses a computer with a mouse and keyboard to navigate files or surf the web.

Besides the novel hardware solutions of the NLS, the system stands out most for its software implementation of a hypertext system that can be seen as a progenitor of the web. The NLS had many of the standard features of a text editor, but stored all text and

content in a series of “statements” which were structured hierarchically (Bardini 136). Arbitrary links could be defined between statements using the “study” command and then navigated using the mouse in an action termed “jumping.” This tool could be used to create a complex web of hyperlinks between any files in the system in a manner very similar to Bush’s Memex (Barini 138). For the first time, hyperlinks were being displayed on a screen and accessed by a mouse interface, although without the graphic signifier that we have come to expect. The mouse worked more like a cursor in a normal command-line based system, still requiring a command to be issued to cause action based on its location. Hyperlinks between statements were displayed with a text-based tag which could be hidden or shown on demand with a text command (Engelbart). The hyperlink was not attached to a particular word or phrase, but to an entire statement (of arbitrary length), and could either move the user progressively through a hierarchy or to an arbitrary, predefined point in any file. NLS demonstrated a novel, hybrid system of hyperlink semiosis: keeping links invisible by default until invoked by a command, then allowing direct selection via mouse.

The text mode of the NLS only expressed uppercase letters, and overlines were used to indicate that letters should be interpreted as capitalized. This overlining does not seem to have been overloaded with meaning, for instance as a signifier of a hyperlink, but since links were attached to statements generally rather than given words, this would not have been considered. In displaying graphics, there is evidence that underlines were used, but to provide emphasis only (Engelbart). While not the origin of the underline in hypertext, the NLS was the first system to use many of the critical components of what

were considered the modern web-type interface, including the mouse, the hyperlink, embedded images, and real time interactivity.

The innovations of NLS had a long-lasting influence on the development of personal computer interface technology and on hypermedia. Nearby to SRI, Xerox established its own research and development group in 1970 called PARC (short for Palo Alto Research Center). Xerox intended to compete in the burgeoning computer market and hoped a West Coast branch could tap into the innovation of Silicon Valley. PARC developed a number of important technologies, including the Ethernet networking protocol and the laser printer, as well as advanced user interface technologies. Based off of the work of Douglas Engelbart, PARC created the Alto computer in 1972, which it released in 1973 (Hiltzik 200). The Alto personal computer used a menu-based display interface and was the first commercial computer to use a mouse. While not commercially successful, its software, especially the Bravo text editor, popularized the interaction methods which would come to dominate the desktop computer space and would be vital in the development of other mouse-based hypertext systems and eventually the web. Most directly, the Alto (and it successor, the Star) influenced the design of Apple Inc.'s (Apple) Macintosh computer (Hiltzik 361).

The Macintosh, released by Apple under the control of Steve Jobs in 1984, *was* a commercial success unlike the Alto, taking many of the novel innovations of PARC and reducing their production price sufficiently to reach the personal computer market. While I will not dwell on the history of Apple, the Macintosh is important to the history of the web and hypermedia for a few reasons. First, Macintosh popularized the direct-manipulation paradigm that is the basis of web interaction (Gladwell). Second, one of the

most widely distributed hypertext programs, HyperCard, was developed on the Macintosh, which I will discuss later. Finally, the Macintosh was the predecessor to the NeXT Computer which was developed by NeXT, Inc., the company Steve Jobs created after leaving Apple. The NeXT Computer was instrumental to the creation of the web by Tim Berners-Lee, to be discussed later.

The impact of the Macintosh on the graphical desktop market cannot be understated, but it was also competing against less expensive computers, including the incredibly popular, but text-based IBM PC. The limitations and capabilities of the display system of a computer have very important implications in the methods chosen to represent interfaces and specifically hyperlinks. The choices made by designers of how to implement a concept visually is as much based on how they *can* do it as much as how they *should* do it. In the following section, I will illustrate the role of path dependency of display technology on the presentation of hypertext systems through an analysis of the affordances of the various display adapters available around this time.

Display Adapters

A computer is useless if it cannot provide functional output to its user, but the methods of doing so have evolved significantly over the history of computing. Early computers used punched card or tape, lights, or mechanical indicators to record output. These were supplanted by teletype-based systems in the 1960s. Cathode ray tube (CRT)-based systems were developed for specialty purposes but did not find widespread adoption until the 1970s. For instance, the military-inspired projects such as Whirlwind at MIT in the 1950s and SAGE used CRT displays in conjunction with a light pen to track

user input, however the high cost of these systems were uneconomical for wider application.

The expense of using CRTs did not stem from the tube itself, as they were readily available from the TV industry. Instead, the expense was related to the memory and control circuitry needed to operate them. The CRT operates by using a signal to modulate an electron beam that scans along the phosphorus-coated screen. In television, the signal is generated from a camera at the point of broadcast, and is fed from an optical input (from a live broadcast of a recording). Due to the limitations of human continuity of vision and the limited time an unexcited phosphor will glow, the screen needs to be swept multiple times every second–roughly 15 times per second in the NLS system–to produce a flickerless image (Engelbart). This does not present a problem for TV, which has as input a continuous signal from a broadcast, and does not need to buffer that input in any way, just convert the signal to instructions for the electron gun. Conversely, a computer has no implicit visual representation of its content; programs only issue output commands sporadically. To replicate the action of a teletype input, simply instructing the CRT to output a message once, would result on a barely perceptible flash on the screen. Instead, the message would need to be repeatedly output many times per second to produce a continuous image, the task of which being so onerous that it would monopolize the processing power of computers at the time. To solve this problem, display buffers were developed which would allow offloading the maintenance of CRT output from the computer itself.

In early screen-oriented computers, developers were forced to compromise between competing needs due to the limitations of the existing technologies, resulting in

some sub-optimal decisions when appraised from an aesthetic standpoint. The special challenge with using a CRT was the technical complexity of the buffer—the hardware used to generate an image. There were two main approaches to output buffers: raster and text-based. Raster outputs store a value for each pixel of the screen in memory. This requires a number of bits for each pixel (this number is dependent on the number of colors available for each pixel), which was very costly in the early days of interactive computing. For instance, on the NLS, the full pixel buffer that would be needed to assign a color individually to each pixel would have cost $15,000 to $20,000, making it an unaffordable choice (Bardini 131). The advantage of such a system, called "bitmapped," however, was the program running on the computer would have complete control over every pixel on the screen independently. A bitmapped display leaves all design choices to the program designer, and their interface is much less beholden to the typefaces or graphics modes of the hardware. The alternative was to use a text-based system which, instead of storing data for every pixel, stored a datum for each cell of a grid into which the monitor was divided. Each cell could be filled with one of a predefined number of characters, for instance the ASCII character set. The pixels to store these characters were stored in hardware and drawn as needed. This significantly lowered the memory needed to store the current display in a buffer, but at the expense of expressiveness of visual display. The choice, and subsequent limitations, of one or the other of these systems had profound effects on the type of display systems and software developed on each system. Also, within each display mode, the nuances of the character sets and options available also constrained output. A developer, when presented with the choice between display modes would choose the one which best served his needs and would have to accept the

tradeoffs entailed. What follows is a discussion of some the most influential display adapters and an analysis of how their limitations affected the software and especially hypertext systems developed on them.

ANSI Terminal/DEC VT100. Before desktop computing overtook timesharing as the dominant paradigm, terminals were used to interact with distant computers. Many standards existed, but the Digital Equipment Corporation VT100 eventually gained enough market share to become a de facto standard, and its code system was adopted by other manufacturers. The terminal used a set of commands called escape codes to allow the computer to control the visual aspects of the textual display. These codes were eventually consolidated as the Select Graphic Rendition section of Standard ECMA-48 which standardized control functions for coded character sets (ECMA 40). These codes were invisible to terminal-users but instructed the terminal how to display the text that was to follow. These parameters included underline, italics, blinking, bold, and others. While this standard grew out of the specific abilities of the VT100, it came to be incorporated in subsequent computer technologies such as the BASH shell, a command line program used in UNIX and its clones and descendants–including Linux and Apple OSX (Loison). Interestingly, the ability for the VT100 to display the underline was not adopted by many of the other display standards that replaced it. Being a monochrome system with fixed-width fonts, the VT100 terminal did not require many bits per character in a frame buffer, unlike the systems that followed, allowing a relatively more advanced capacity to format characters. It would take a number of years for these to be standard on desktop computers, although the capabilities included in ECMA-48 would

shape early text-based browsers such as Lynx, which I will discuss in the following chapter.

IBM Monochrome Display Adapter (MDA). The IBM PC, released in 1981, was an immediate success among business users who were seeking inexpensive personal computers. Unlike the Apple Macintosh, released in 1984, which used a graphic display, the PC used a monochrome, text-based display. The display was 80 by 25 characters in size and supported 256 different characters codes in its 8 kilobyte character generator. Each character *box* was 9 by 14 pixels, while the characters are represented in a grid of 7 by 9 pixels (IBM 1-123). This box to character ratio is indicative of the manners in which characters can be modified. Each character on the display was represented in the display adapter as a pair of two bytes: one byte (8 bits) for the character itself (of the 256 available), and one as an attribute code. Three bits each of the attribute code controlled the display of the foreground and background of the character. One bit set the given character to blink, and one bit set the intensity of the character. These codes could be combined to hide the text, reverse the video or underline the text (IBM 1-126).

Two attributes of the design of the MDA allowed the possibility of using underlining on the IBM PC: the use of a character box with sufficient spacing to fit underline pixels, and the choice to reserve one bit of the attribute code dedicated to represent the underline. Especially at the lower end of the personal computer market, memory was at a premium and only a finite amount of metadata about each character could be stored without adding more memory. Other methods of modifying text would require other tradeoffs; adding italics could have been achieved by increasing the

memory of the character generator, adding italic characters, and increasing the width of the character code. But whereas an underline is combinatory, italics would require an entirely different character set and associated memory to store it. In many ways, the MDA mimics the capabilities of desktop printers (in fact, the full name of the adapter is the "IBM Monochrome Display and Printer Adapter") and also controlled a printer attached via a parallel connector (IBM 1-117). Because of the lack of expressiveness in the formatting characters on the IBM PC, engineers developed workarounds to ameliorate the monotony of plain text in the form of "code page 437" which was the default character set included in the graphic adapter. This series of characters included not only the standard numbers, letters, and symbols that would be expected, but also a series of box-drawing characters (Microsoft). These could be combined to form single and double-walled boxes when placed precisely around text and were used to create rudimentary visual user-interface elements such as windows and buttons. By way of contrast, the bitmapped Macintosh used a character set which omitted box-drawing characters in favor of various symbols (Connolly 60). As a result, application designers on the IBM PC made extensive use of these characters in their applications not necessarily because they were optimal—the boxes allowed for at best ersatz visual user interfaces—but because they were available.

IBM Color/Graphics Monitor Adapter (CGA). Unlike the MDA, which also controlled printer output, the IBM Color/Graphics Monitor adapter was only meant to control video output to a monitor or television set (IBM 1-133). The CGA had two modes of operation: first, a text mode called A/N (alphanumeric) similar to the MDA. This mode supported a

40 by 25 grid or 80 by 25 grid (on high resolution monitors) with a box size of 8 by 8 pixels. Each character with a box was 7 by 7 pixels, not leaving space for an underline. Like the MDA, each character in the grid was represented by two bytes, one representing one of 256 characters, the other for attributes (IBM 1-133). The need to represent color information required a change in how the bits in the attribute byte were allocated. Colors of the fore- and background were independently selectable on a per-character basis, from a list of 16 colors (8 colors, each in a low- and high-intensity version) (IBM 2 6). To make room for this, the underline bit was dropped from the attribute byte forcing developers to choose one of the other methods to emphasize characters, for example color or inverse video.

The CGA also allowed for Graphics Mode in which specific pixels were individually addressable. Three modes were available which traded off between number of pixels and number of colors per pixel (IBM 2 9). Underline could be used in this mode (as could any other interface he wanted to design) if the designer of an application chose to include it, but a developer of business applications (for example, Microsoft Word, in its original incarnation) would have been more likely to develop in the A/N mode, using box-drawing characters where necessary to simulate a graphical interface (IBM 1-114).

IBM Enhanced Graphics Adapter and Video Graphics Adapter. IBM improved on CGA and released the Enhanced Graphics Adapter in 1984, but the next major revision was Video Graphics Adapter (VGA) in 1987 (Hart). VGA also supported A/N and graphics modes, with an expanded list of supported colors and resolutions. While in A/N mode, the same bit scheme was used as in CGA, however the focus on the card, as the name

implies, was more on the graphic display modes, of which it had six (IBM 3 2-1). By this point, with the success of Apple's Macintosh and the growing popularity of Microsoft Windows, it was clear that graphical displays were becoming the standard. The lack of underlying support in A/N mode was made less relevant by the increased reliance on graphical display modes to generate interfaces and text.

Apple Macintosh. Unlike its earlier forays into personal computers, such as the Apple II, the Apple Macintosh was envisioned as a purely visual, desktop-metaphor based machine with an integrated nine-inch screen of 512 by 342 pixels (Connolly 18). Unlike other popular display adapters in the 1980s, the Macintosh relied solely on its bitmapped display mode, eschewing the then-standard text mode. Apple developed software called QuickDraw[5] to handle the rendering of the complex interface and manage screen buffers (Connolly and Lieberman 84). This technology removed the limitations of how text could be styled from the hardware layer: a programmer could make software that could display any type of modified fonts, as indicated by the many styles available out of the box, including bold, italic, underline, outline and shadow (Connolly 114). These graphics modes were built into the operating system layer which tightly controlled the look and feel of applications. In the same way that hardware graphics modes offered "building blocks" to developers, Macintosh offered these tools on the software level.

A notable omission from this section is the NeXT computer, the machine on which Tim Berners-Lee wrote the original WorldWideWeb software. While closely

5. Not coincidently, the developer of QuickDraw, Bill Atkinson also developed HyperCard, a hypertext system to be discussed later in this chapter (Aker 98).

related to the Macintosh, I will, instead, discuss the machine in the next chapter, as it is so closely tied to the web's development.

Authors of web history, including Jakob Nielsen and Robert Cailliau have published excellent scholarship on web history and usability, but do not sufficiently consider the material constraints of the hardware on which the early web was built. This chapter has attempted to extend their work by including the details of hardware in the analysis. To this end, I have chosen to go into the details of the hardware of these early display adapters to illustrate the environment in which programmers were developing their software. These programmers were not presented with a *tabula rasa* to create any software they could imagine; instead, they had available to them a set of tools which could be used to assemble a product which would achieve their goals. I do not mean to describe these affordances in the negative: as the artist Matthew Barney's *oeuvre*, "Drawing Restraint" demonstrates, from constraint can come great works of art. However, I do posit that the question of choice may never have even risen to the level of consciousness, in some cases. For example, when presented with the need to highlight text on a platform in which using inverse-video is possible and underlining is not, a programmer is liable to use the former without ruing the absence of the latter. The material constraints of the hardware had agency, affecting the outcome. The affordances of each machine shaped the software created on each one. This principle is at once obvious and vitally important to understanding the development of the early visual presentation of the web. This chapter has only cursorily discussed the cost of hardware,

but suffice it to say that for equivalent devices costs were multiple factors of what we would pay today, even accounting for inflation[6].

This chapter has provided a prologue to the invention of the web and the history of the uses of the underline and the visual display of hyperlinks as well as an overview of the hardware on which early web browsers were developed. All of these topics are vitally important to understand what choices were consciously made versus what were determined by existing technology. It is often difficult to disentangle these factors, but a good understanding of the technological milieu can help elucidate the issue. This is especially important when viewed decades later once radical advances in technology have obliterated many of the constraints from the time. The stage is set, as it were, for the birth of the web.

6. The archives of popular computer magazines such as PC Magazine and Macworld provide a good source of contemporary pricing and availability of hardware.

Chapter III

The Invention of the Web and Underline as Hyperlink Semaphore

The web was not developed in a vacuum, but amid of a great deal of research around hypertext and hypermedia systems in both academia and industry. In the late 1980s and early 1990s, most of this research was focused on offline systems due to the dearth of networked computers in consumer and business environments. As an incipient field, many of the human-computer interactive aspects of hypertext systems were in flux, and designers of systems such as HyperCard (and eventually the web) experimented with a variety of modes to display and interact with hypertexts. These offline systems, and the early networked systems, shaped the understanding of hypertext and ultimately affected the development of the web. The discussion below focuses primarily on how these systems represented hyperlinks in order to demonstrate the contemporary environment in which Tim Berners-Lee developed the web.

Contemporary Hypertext Systems

A number of hypertext systems achieved some level of success before the web. The systems below all made an important impact on the public perception of hypertext and its visual implementation.

HyperCard

Released by Apple Inc. in 1987, HyperCard was the first hypertext system to be widely adopted by a general audience. For many users HyperCard was their first

introduction to hypertext, whether they understood it by that name or not. The novel concept of hypertext, coupled with the power and flexibility of the HyperCard software package, presented a communications challenge to Apple because unlike most software, HyperCard did nothing on its own. Instead, it was a chimera: it could be used by both developers and users to create their own applications, including hypertexts. The software presented a visual canvas to position GUI elements coupled with a database model and a scripting language (HyperTalk[7]) to facilitate interactivity (Vaughan 3). The rapid spread of HyperCard among the non-specialist population, as well as its ease of use and ability to create hyperlinks, vastly expanded the knowledge of–and expectations for–hypertext systems in the late 1980s. The hypertextual capacity of the software even influenced its name–originally dubbed Wildcard, owing to its multifaceted abilities–but there is evidence that HyperCard also influenced the public perception of hypertext. Vaughan describes the process by which it was rechristened HyperCard: "Some people say the Hyper prefix was tacked on to associate the product with the cachet of the emergently fashionable hypertext phenomenon. Yet the chronology also suggests HyperCard's arrival is largely responsible for hypertext's new status as a fashionable idea" (Vaughan 25).

As discussed in the previous chapter, many earlier hypertext systems existed, but they were meant for specialist audiences. Apple capitalized on the widespread adoption of affordable personal computers and the burgeoning market for all types of software, including hypertext systems, by choosing to include HyperText with every Macintosh computer beginning in 1987. The author of HyperCard, Bill Atkinson, also created the

7. Apple initially was not going to include HyperTalk language, thinking users would only use the editing mode to make stacks, but upon seeing its potential for extensibility, decided to include it in the distribution (Vaughan 3)

digital graphics program, MacPaint (which was also bundled with the Macintosh), and described HyperCard as the "lowest common denominator" (Vaughan 30); by including it on every Macintosh, the audience for HyperCard *stacks* (as programs created in HyperCard were called) would be extensive. The influences of Atkinson's MacPaint are visible in the toolkit included in HyperCard, allowing extensive screen painting capabilities and facilitating graphically-oriented user interfaces in an era when text-based interaction was considered standard. Cutting-edge for its time, the preference for graphical interaction in the included software reinforced the Macintosh paradigm of visual over textual interfaces. Despite HyperCard's influence, the paradigm of using a distinct link layer separated from content was an evolutionary dead end in hypertext systems.

While HyperCard was certainly influential in spreading the concept of hypertext to a general audience, it was not to be the forerunner of the web due to its single-user nature. Stacks existed both for personal software (such as contact lists) and software meant for broad distribution (such as encyclopedia-type software) released both on physical media and on the internet. Vaughan reports that in 1987 over three hundred stacks were available on Compuserve, but while widespread, each stack file could only be used by one person at a time (35). The HyperTalk protocol allowed for network interaction through extensions, but at its core the stack was meant for the solitary desktop user to interact with hermeneutically.

The predilection for visually-oriented rather than textually-oriented hyperlinks was baked into the structure of the HyperCard design system. HyperCard used the paradigm of the notecard to structure and present any type of data. Links could be

established between cards with user-defined hyperlinks, or through scripting with the HyperTalk language, which was extensible for many other purposes as well (Vaughan 16). Each card consisted of a graphical background with an overlay of other controls, including links. Individual words or paragraphs could not be be directly linked (in contrast to systems from decades earlier like NLS), although a similar effect could be achieved with some effort by using a series of precisely placed invisible links on top of the desired word. The omission of textual linking makes clear that the intention of the developers was for the visual paradigm (buttons) to supercede the textual (inline links).

Early HyperCard stacks functioned as simulacra to aid their adoption using *trompe-l'œil* techniques (using artistic tricks to make flat images appear to have depth) to replicate the visual aspects of a physical referent. As Baudrillard describes in *Simulacra and Simulation*, the precession of the simulacra begins by representing the physical world as a simulation. Over time, this simulation becomes disjoined from reality and operates as its own *hyperreality,* having lost any reference to the original object and becoming its own reality. An advanced Rolodex stack purports to be a "Rolodex," and has *become* a Rolodex to anyone not familiar with the real thing. This process can be observed with many computer applications, especially early Hyperlink stacks. As an example, the earliest successful stacks were digital Rolodexes, storing contact information in an easily searchable and linkable format (Vaughan 43). The earliest examples of successful stacks were faithful simulations of the real-world object, but over time additional features added until they became an entity unto themselves. Digital contact lists have, to a great extent, rendered Rolodexes obsolete, so the conversion to a true simulacra–in the Baudrillardian sense–is almost complete.

The Rolodex-clones used graphical depictions of paper cards to display contacts, and superimposed icons (using symbols such as arrows or magnifying glasses) to function as links. Inline hyperlinks were not in circulation as hyperlink signifiers at this time, primarily due to the complexity of implementing them. An icon could be imbued with a "listener" that would respond to a click by activating a predefined HyperTalk script. The "trick" of using invisible links could be employed to insinuate a more complex interface, but centering a link on an individual word was complex in that editing the text at all would require a manual shifting of all links. This anti-affordance ensured that inline hyperlinks were never adopted in HyperCard. Were the capability to link text inline to exist, it is very possible that the underline could have been used. Indeed, underlining was available in HyperCard using its WYSIWYG text editor. Underlining could also be achieved programmatically using HyperTalk, which included a function–"textStyle"–that allowed various types of formatting to be applied to any text on a card, including text in a field, on a button, or on the background (Vaughan 601). The two functions were never combined, however.

As the discussion of HyperCard demonstrates, the web was not the first instance of a hypertext system available to users. The web was also not the first widely used graphical information system to use links, although soon after its launch the web decimated its contemporaries. The web succeeded where others experienced only limited adoption for a number of reasons, discussed below, but its particular mode of representing hyperlinks helped speed its adoption. It is worth discussing these systems briefly to highlight the way in which the web was revolutionary and flourished. This discussion will also highlight the alternative modes of representing links available to

developers at the time, and show that Berners-Lee's use of the underline was not preordained.

Minitel

The French government, through its telecom monopoly, PTT, developed a videotex service called Télétel with an associated hardware terminal, Minitel[8], to provide visual-textual information to subscribers over their phone lines. First deployed in 1978, subscribers were issued a free-of-charge terminal with integrated keyboard, monitor, and modem which could create connections to various services, some free, some for a per-minute charge. Services proliferated through the 1980s and included shopping, financial quotes, and news, with subscription rates skyrocketing from 11,000 in 1982, to over half a million in 1984, to three million in 1987 (Abadie 108). While the web was beginning to blossom, Minitel continued to boom to six million users by 1993, bolstered by Minitel's killer apps: a free-to-use nationwide telephone directory and classified ads (Cats-Baril and Jelassi 1). However, the mid-1990s were the peak of the Minitel's success with approximately nine million users, and it was supplanted by other services, including the web. The service died a slow death, with holdouts in certain industries–such as farming–providing pressure to keep it active until its decommissioning in 2013 (Sayare).

Like other terminal-based applications, the Minitel console was limited in its ability to represent visual information, both by the hardware of its monitor and the speed of its connection. The monitor used an "alpha-mosaic" display scheme: each character was addressable and designers could represent a letter, number, symbol, or colored

8. The total system will be henceforth referred to as Minitel, per common usage

blocks to simulate graphics, with each cell acting as a crude pixel (Kessler). Available commands were indicated on the Minitel screen in various schemes, but many used tables to connect commands to the codes users needed to type to follow the link. Lacking a mouse, all commands were entered using these codes or shortcuts through the keyboard, making the system almost menu-based. Unlike the web–especially as initially envisioned by Berners-Lee–Minitel was intended primarily as a distribution channel, with the vast majority of data flow being from publishers to users. This is demonstrated by the asymmetrical modem hardware packaged with the original terminal: the upload speed was much more limited than the download speed, as input was only envisioned to be piped from the keyboard rather than uploading any types of files (Kessler). While its adoption was limited to France[9], primarily due to the government's reluctance to expand with a more open network, Minitel was a significant advancement in network computer based, visual data services. However, the Internet and the web in particular, owing it their open and free form protocols, eventually supplanted Minitel, albeit later in France than in other European countries. The textual, code-based interface of Minitel was an advancement over the many paper-based systems it supplanted (telephone books, classified advertisements, agricultural reports, etc.) but its interface was still limited to keyboard command, with minimal interactivity by design. Others have argued that the installed base of Minitel coupled with its "killer apps" allowed it to keep running beyond its time, but it is worthwhile to look at the question in the reverse: why did Minitel not expand beyond France? Certainly, government policy was important, but if sufficient demand existed in other countries, it could have been exported (there was a test in

9. Other teletex systems were established, but none has the market share of Minitel.

Belgium), or other national Teletex systems would have taken off; many other systems were launched, but none succeeded like France's. The lack of success might be partially attributed to the competition from the web due to its more appealing and usable interface: at the time when Minitel was reaching its highest usage, the web was beginning to take off and Mintel's dated interface could not compete.

Gopher

Probably no technology had the potential to take advantage of the same latent desire and technological milieu that allowed for spread of the web than Gopher. The interest in hypertext systems and the expansion of the internet provided fertile ground for the development of such online communications systems. The study of Gopher is important because it presents a symmetrical counterexample to the web: a similar product, created in a similar social-technical milieu which did not succeed (Pinch and Bijker 406). Ogburn and Thomas wrote about this effect in 1922 in their essay, "Are Inventions Inevitable?" and identified that with regular frequency major inventions are simultaneously co-invented. They attribute this primarily to what they call "existing status of culture:" potential inventors, possessing "considerable mental ability," all are privy to roughly the same knowledge and techniques to build upon at the same time(86). To paraphrase Isaac Newton, they were all standing on the shoulders of the same giants. Were it not for some small difference at key times, it is conceivable that Gopher could have achieved ascendence rather than the web.

Gopher was created at the University of Minnesota in 1991 by a team led by Mark P. McCahill to improve the computer center's help system, and eventually grew to be a

full campus information system (Gilster 37). Like the web, Gopher used a client-server architecture to display information on screen, although it was limited to textual content. Despite this limitation, the system spread from the University of Minnesota to other academic, research, and commercial institutions such as the University of Texas, The Electronic Frontier Foundation, and NASA (Gilster 45).

While a great deal of content was available on "gopherspace," the menu structure was simplistic: all pages were arranged in a strict hierarchy. On each page, the user was presented with a menu of available sub-items. Selecting one of the items would bring the user either to another menu, or to a page of content; alternately, the back command would take the user to the parent page in the hierarchy. The menu structure allowed a type of hypertext–pages could be read non-sequentially–but not a hypertext system as robust and intuitive as the web. Visually, there was no dedicated style for links, because all links were part of a menu. The menu lists were numbered, featuring an arrow on the left side to indicate which menu item was currently selected. Because Gopher was intended to be accessible either by terminal or browser software, no graphical capabilities were assumed and thus styling was minimal (Gilster 42).

At the time, much of the academic conversation around hypertext was focused on the question of how best to demonstrate the underlying structure of the hypertext. Nielsen described systems such as Perseus, which rendered content nodes as pictures of footprints on a map to present a metaphor of a tourist visiting the content, or the use of overview diagrams to visualize the connections between nodes (128-133). The only free-form linking available to users of Gopher was the bookmarking system. Users could create a personal bookmark list that was accessible from any point in the system and would

immediately take them to any page in Gopherspace, or to a predefined search string (Gilster 56). However, the fact that these bookmarks were created by the end user rather than the content creator made it a fundamentally different type of hypertext system than the web. In hindsight, the web succeeded because of, not despite, its lack of enforced hierarchy or top-down presentation of the hypertext.

Melvin Conway coined an eponymous law about the way software is shaped by the organization that develops it: essentially, the cultural context in which a software product is created can have a profound effect on the timbre of the system. Gopher, a university campus information system, was intended to distribute information in a top-down manner, and the system that was developed replicated the hierarchical organization structure of the university system. In contrast, the web was developed at a sprawling research institution with a focus on collaboration, and was, not surprisingly, more anarchic in nature.

Despite Gopher's initial success, it was quickly supplanted by the web. The web has many benefits over Gopher: integrated graphics (on supported browsers), lack of top-down structure, free-form hyperlinking. Tim Berners-Lee, however, posits that the main reason that Gopher did not succeed was its financial model. In 1993 the University of Minnesota announced its intention to charge an annual fee to license its Gopher server software to non-educational entities (Berners-Lee and Fischetti 72). Whereas previous internet-based communication systems, especially those born of the university, tended to be free to use, the commercialization of Gopher stymied its development. Berners-Lee might have been demuring when he cited the commercialization of Gopher as the cause of its downfall; the web was also beginning to blossom around the same time, and thus

stole much of its market share. The web could do everything that Gopher could, including accessing Gopher (with dedicated gopher:// prefix) and more, and without any licensing fee–a stipulation that Berners-Lee thought critical to allow for its success, thus releasing his project under the GPL or General Public License (73).

NeXTSTEP

This chapter has so far described the hypertextual milieu in which Tim Berners-Lee existed when creating the web. Before discussing the actual nascent stages of the web, it is worth considering the specific technology on which he accomplished this development. This section will switch from a ten-thousand foot view of late 1980s hypertext to the microscopic view of Berners-Lee's desktop environment. The tools available to him both enabled and constrained his development in ways that shaped the future of the visual presentation of links on the web. Had he developed on other common hardware, such as the previously mentioned Macintosh, the resultant reference software and specification would likely have been very different. The following discussion of the hardware, software, and ecosystem of the NeXT computer will show the influence it had on the design of the web.

While working at CERN in 1989, Tim Berners-Lee used the new, untested computer called the NeXT Cube to develop the first web browser. The Cube was expensive compared to other desktops available at the time, and it was used by few of his colleagues. However, the graphical capabilities and object oriented design of the Cube made it consonant with his vision, and also shaped the visual presentation of the first web browser, WorldWideWeb. The Cube was released in 1989 by NeXT Inc. (which later

become NeXT Computers) and was the first model ever produced by the company. The history of the NeXT is tied up with that of an earlier computer referenced previously, the Apple Macintosh. Steve Jobs, one of the founders of Apple, left the company in 1985 to start NeXT. NeXT intended to produce an advanced desktop/mainframe hybrid for the higher-education market which, Jobs claimed, would not directly compete against Apple (Stross 74). The claim proved to be disingenuous, as NeXT eventually began marketing the computer to other industries as well, to the chagrin of Apple which brought suit seeking an injunction.

Jobs was unable to replicate his success at Apple with NeXT. The computers they produced, while being well-designed and featuring a very advanced operating system, were deemed too expensive for their market and the expected demand for them never materialized. Many of the technologies used in the NeXT, such as voice email, could only be taken advantage of if many other users also had NeXT computers. Network effect can be a very powerful force towards adoption, but NeXT's focus on one industry and its high price point were too strong of counterforces. Stross put the relative popularity of the NeXT and the Macintosh in perspective by pointing out that between 1985 and 1992, NeXT sold 50,000 computers, while Apple sold that many in the average three day period in 1993 (3). The lack of adoption of NeXT at CERN was even a stumbling block when Berners-Lee was trying to introduce the web to internal users there, as will be discussed in the next section.

From the beginning of its design, the NeXT was intended as a revolutionary, visually oriented computer, much like the Macintosh was in 1984. Like the Macintosh, the NeXT Cube was sold as a package with a special mouse, keyboard, and monitor

designed for and required by the computer; a NeXT laser printer was optional. A great deal of effort was put into the visual output of the computer, both on the screen and the laser printer–to the detriment of the actual power of the machine, according to its critics. The core visual technology of the NeXT was Display PostScript, a version of the PostScript language which controlled the display of images on the screen and on the printer, resulting in true WYSIWYG output (Stross 12). The Display PostScript software ran on top of Window Server, which in turn ran on the Mach Unix kernel, the basic operating system of the NeXT (Webster 42). The NeXT monitor, called the MegaPixel display, was a 17" CRT coupled with a 256k video buffer capable of displaying 4 levels of grey at a resolution of 1120 by 832 pixels (Webster 38). The monitor was large for the time and its 94-dpi resolution was very clear, but color screens were becoming more common so it was not received as such a revolution as the Macintosh (Webster 41). The computer hardware itself was competitive, but not exceptional for its time and did not wow consumers. Its computing hardware was built onto one robot-assembled card which connected to other internals including a magneto-optical drive and optional hard drive through an internal communications bus: this bus also allowed the insertion of hypothetical expansion cards which never came to fruition (Webster 12). Finally, the NeXT came with a built-in port for ethernet and the software to support communications over the protocol. Much of the appeal was in its visual design–a striking, jet-black cube–its provenance–being the newest computer from Steve Jobs–and its object oriented software system–NeXTSTEP.

Rather than using an existing architecture, NeXT Inc. opted to use a new CISC processor from Motorola. This decision rendered the vast array of programs available on

Macintosh and IBM machines out of reach of the NeXT Cube. Any existing software would have to be rewritten: a task that was not appealing to software companies for whom an investment in re-architecting their products was a matter of speculation predicated on the widescale adoption of the NeXT. As a result, the computer launched with almost no third-party applications (Stross 169). Not surprisingly, much of the popular press lambasted NeXT for the deficient software library. However, Bruce F. Webster, in *The NeXT Book*, a 1989 user guide and introduction to the platform, spun it as a selling point. He claimed that the Interface Builder software included with the NeXT was sufficiently easy to use and powerful that users would *prefer* to build their own software to suit their needs rather than purchase pre-built software off the shelf (99). Whether or not this was a viable marketing strategy, Interface Builder did allow Tim Berners-Lee to develop his reference browser without the need for additional tools or development environments.

The software that was prepackaged with the NeXT Cube was rather advanced and took advantage of the Display PostScript subsystem to facilitate well-designed, easy to use software, such as the WYSIWYG WriteNow text editor. NeXT programs were created using a series of prepackaged utilities. At the highest level, Interface Builder gave developers tools to quickly create graphical user interfaces (GUIs) with precoded widgets (Webster 128). Using standardized drag-and-drop components, the tedious task of programming basic interface elements and their associated handlers was streamlined, speeding development time and increasing standardization amongst software. Importantly, the Interface Builder output its code in Objective-C; while other languages could be used in NeXT, the integration of the Interface Builder biased application

developers to use it. The Application Kit software library formed the foundation for the Interface Builder by providing the needed functions for developers to interact with the lower levels of the system, including Window Server which handled screen drawing tasks (Webster 119). This powerful software stack afforded developers the ability to create fully-functional programs using the Objective-C language which could take advantage of all of the hardware capabilities of the NeXT.

As the name implies, Objective-C is an object-oriented language based on C; this dovetailed with the NeXTSTEP system which itself was object-oriented, a novel concept in commercial systems (Webster 131). This method of design allowed for extensive reuse of code among applications and simplified application development. The NeXTSTEP system was actually the NeXT product with the longest influence, eventually being ported to run on standard computer architectures after NeXT abandoned hardware production and sold its intellectual property to Apple (Stross 177). The use of the object-oriented techniques is visible in the design of WorldWideWeb, the first web browser, which Berners-Lee developed on the NeXT Cube. Most of its interface, what we now call browser "chrome," is built with stock components of the NeXT operating system. The menu structure in NeXTSTEP was a standard component across applications and included submenu panels such as Document, Edit, Find, and Font (Garfinkel 32). The font menu includes a limited number of text styling options that recur throughout the system such as bold, italic, and underline. Programs could implement these menus as-is or add or remove items from them in their programs. WriteNow, for instance, strangely chose to omit underline as an available option for formatting documents (Webster 242). NeXTSTEP programs could also use Rich Text Format (RTF), a markup language

created by Microsoft to allow extensive formatting of display documents in code (Garfinkel 311). Due to its fluidity with presenting material on the screen and in print (owing to its Display PostScript software) and the demand for desktop publishing, the NeXT was designed with extensive text formatting capabilities, giving designers great flexibility in displaying text.

It is interesting to note that the NeXTSTEP system came prepackaged with an existing hypertext in the form of its online help system. The help system could be accessed from most application menus, and individual articles were cross-linked with others. Links were listed at the bottom of articles and signified with a small diamond icon next to the link title. The inclusion of a hypertext system in this context demonstrates that, by 1989, offline hypertext was already commercially applicable and the paradigm was primed for wider adoption. The NeXT computer, with its advanced conceptual underpinnings, was an obvious candidate for the development of a revolutionary system like the web, but it also made it ahead-of-its-time for a wider user base.

The capabilities of the NeXT computer and NeXTSTEP led Tim Berners-Lee to choose it to build his demonstration web browser. Many of the particular affordances of the computer platform then influenced his actual implementation of WorldWideWeb. The use of Objective-C to code the browser was a product of Interface Builder and its object-oriented code creation. NeXT's ethernet capabilities were a prerequisite for the development of HTTP over the CERN network. Finally, the windowed graphical capabilities of HTML would have been difficult to implement on an alternative system. WorldWideWeb could have been developed on another platform, such as X Window or

Macintosh, but the outcome could have been very different due to the best-practices and stock functionality of those systems.

Tim Berners-Lee and the World Wide Web Standards

As the inventor of the web, Tim Berners-Lee's life and work has been well documented, and I will not attempt to summarize it extensively here, however it is worth noting that when one attributes the invention of the web to him, it is not hyperbole. His work on creating the three fundamental components to the web–Hypertext Markup Language (HTML), the Hypertext Transportation Protocol (HTTP), Universal Resource Locators (URLs[10])–and their corollary the web browser (WorldWideWeb), was not the product of a committee or extensive collaboration. Each of these aspects of the web have technological precedents, of course, having been based on some underlying technology. In fact, Ralph C. Epstein debunked the "heroic theory" of invention in favor of a "theory of small increments" as far back as 1926 (237). By observing that "a given invention is merely the culmination of a train of effort reaching back into the past and brought to fruition in the present" he asserted that "no invention is dependent upon the presence of any one individual" (242-246). This section makes no claim counter to Epstein's observations, however, approaching the development of the web by looking at Berners-Lee's decision is appropriate in this case because of his role in furthering each technology and combining them into a new whole. Of course, once it was released to the world, the web took on a life of its own, all the more so due to its open standards and lack of

10. The name Universal Resource Locator was eventually changed to Universal Resource Indicator to acknowledge the fact that the location of files and page could change over time. For the remainder of the paper, URI will be used.

commercialization. In turn, this section will discuss each of the three web standards Berners-Lee developed and how they shaped the development of the underline being used as a hyperlink signifier.

Tim Berners-Lee worked at CERN, the European high-energy physics lab, as a programmer in the 1980s, first as a consultant and later as a fellow (12). Despite taking a degree in physics, his task was not directly physics-related, nor was he hired to write a hypertext system. Berners-Lee worked on the more prosaic work of networking and software development. Both of his first hypertext projects were side projects: Enquire, written around 1980, contained the kernel of the web, but was not distributed and also represented links as a list below the content, like many previous systems (17). The method of link lists was also visible in Vannevar Bush's Memex concept, which, along with Ted Nelson and Douglas Engelbart was cited as an influence by Berners-Lee (5). CERN was highly networked but with heterogenous systems that made intercommunication between users difficult. It was from this challenge of sharing knowledge that the impetus of the web was born; in fact, the challenge had been taken up previously without success at the lab, but with attempts that were too structurally restrictive, in Berners-Lee's opinion (15). His proposal to develop a successor to his Enquire hypertext system languished for some time before being approved in 1990, when he was granted funds to buy a NeXT computer and spend time working on his vision (23). This decision to use the NeXT rather than the popular Macintosh or IBM systems commonly used at CERN would prove to shape the visual aspects of his product. Berners-Lee, who was respected at CERN for his brilliance and foresight, posits that his

interest in and enthusiasm for the new computer was a larger factor in receiving approval for the web project than institutional interest in the project itself.

What would become known as the web was named World Wide Web (or W^3) in Berners-Lee's original proposal, and early writings. The first two components of the proposal, HTTP and URIs, were probably the most important, but the least relevant to this study so they will be discussed only in brief. HTTP "is a very simple internet protocol, similar in implementation to FTP and NNTP… [which]… sends a document identifier with or without search words, and the server responds with hypertext or plain text. The protocol runs over TCP" (Berners-Lee, et al. 4). Berners-Lee intended the protocol to be be very simple and easy to implement, and to cause a minimum of overhead to users. The other specification, URI (originally UDI, for Universal Document Identifier) is a standard format to identify documents on the web and was agnostic towards the protocol of the server, so it was not limited to the HTTP protocol. In this way, Berners-Lee hoped that the web could coexist on the internet with other protocols (Berners-Lee and Fischetti 38). These two protocols worked together to allow the web to exist in a fully decentralized fashion, with no central authority, thanks to the URI structure. Berners-Lee cites decentralization as the key reason why the web flourished where other systems such as Gopher did not (Berners-Lee and Fischetti 37).

Berners-Lee developed HTML as a way to structure hypertext and add links between documents in a manner that was lightweight, rejecting advanced formatting capabilities. He cites T_eX as an example of an overly powerful solution to the problem of formatting text within HTML, and raises the "principle of least power" as the reason he chose simplicity over complexity in formatting (Berners-Lee and Fischetti 182). Besides,

advanced formatting could be handled with third-party content handlers. From the beginning, Berners-Lee was concerned with interoperability of the web across multiple computer platforms; he wanted to ensure that all elements of his specifications would work on any popular computer platform–at least to some extent–so limiting the visual options available was a way to create a least common denominator for browsers. It is ironic, then, that the first web browser was a created on the NeXT platform: an uncommon and highly sophisticated computer system. The early web browsers will be discussed in the next section, but the first paper published by Berners-Lee about the web made specific accommodations for browser software of all capabilities, and defined HTML to work across all of them. In his paper, "World Wide Web: An Information infrastructure for high-energy physics," published under the auspices of CERN, Berners-Lee describes the display of hyperlinks in HTML:

> On a graphic terminal, a reference is represented by a sequence of highlighted text, or an icon. The user clicks on it with the mouse, and the referenced document (or part of a document) appears. On a line mode terminal, a reference is represented for example by a number in the text: the user types the number to follow the reference. (Berners-Lee, et al. 1)

His view was that the need for interoperability and increased adoption outweighed graphical presentation. When the paper was presented in 1993, Berners-Lee was still pushing for the wider adoption of the web at CERN, let alone the rest of the physics community and the world.

While Berners-Lee had a vision for the web, its broader meaning was not fixed. Through the lens of social construction of technology (SCOT) as elucidated by Pinch and Bijker, the meaning of a given technology is not implicit to the technology nor monolithic, but shaped by the understanding of the technology in its context (421). The

interpretive flexibility of a technology could be seen in the conflicting view that some users at CERN has of the web. Many viewed it as powerful internal directory: unaware or uninterested in its potential as a global data system. Neither view was correct, but had closure of the web's meaning not been reached as it had (in SCOT terms, the stabilization of the artefact), it could have been pigeonholed into that one use (Pinch and Bijker 424).

HTML

The specification of HTML had an effect on the visual display of hypertext on the web, while also leaving many design decisions up to web browser software. This section discusses the evolution of the specification of HTML and the extent to which it constrained display aspects. While the URI and HTTP protocol were key to making the web what it was, Berners-Lee underestimated the importance of HTML in making it a success. He expected that HTML would be just one option to present content over the web, no more important than any other content system. In time, HTML proved to be wildly successful because of its ease of use, simple code structure, and graphical orientation. To ease the acceptance of HTML at CERN, Berners-Lee chose to model it on the SGML standard, which is a meta-language used to define markup languages (Berners-Lee and Fischetti 41). SGML had adherents at CERN and the familiarity with the structure helped researchers there comprehend what this new system was. As revolutionary as the web was, nothing is created from whole cloth; a system that is too advanced will have a hard time gaining supporters who will have difficulty understanding the system, even if it promises great power. Berners-Lee intended HTML to be both

familiar enough to be readily understood by users, and simple enough to function as a basic interchange format that all servers could, at a minimum, speak and understand.

HTML had been in development since 1990, with implementation information available on the CERN website, but the first *published* HTML proposal was released as an "Internet Draft" of the Internet Engineering Task Force by Tim Berners-Lee and Daniel Connolly in June 1993. This document contained an overview of the web with descriptions of how HTML interacts with HTTP and URI, as well as a full specification of the HTML syntax and tags. For many tag descriptions, the authors included rendering instructions which described "[t]he form of presentation of information to the human reader" as well as noting its *typical* rendering which "is not a mandatory part of the standard but is given as guidance for designers and to help explain the uses for which the elements were intended" (2). While many HTML elements have renderings described, that section is notably absent for anchor (<A>) tags, which define hyperlinks between documents and are therefore one of the most important elements in HTML. The only hint given as to specific rendering of hyperlinks is a note under the METHODS attribute of the anchor tag documentation which "provide[s] information about the functions which the user may perform on an object" (14). It states, "[f]or example, the browser may choose a different rendering as a function of the methods allowed (for example something which is searchable may get a different icon)" (14). This suggests that the visual indication of hyperlinks is left up to the browser to determine, and that icons could be used to do so. The standard proposes the idea that different link relationships could be codified using the LINK attribute of anchors, and that based on the link relationship value could be styled differently. In practice, this was not used extensively, with most anchors

being undifferentiated and thus styled identically. Were this feature widely adopted, the visual presentation of hyperlinks may have been much more multifaceted in practice. However, trying to represent many different types of links presents its own design challenge: Jakob Nielsen tested the ability of users to distinguish link types in the Guide hypertext system and his results, published in *Multimedia and Hypertext*, indicate that, "[i]f the hypertext will have only a few link types, then color or line patterns are suitable choices, but human ability to understand and distinguish such an encoding is limited to about seven different values" (135). This omission of a typical rendering of hyperlinks is interesting in contrast to the detailed manner in which the specification defines other elements. For example, the Address tag's typical rendering is defined as:

> Typically, an address element is italic and/or right justified or indented. The address element implies a paragraph break. Paragraph marks within the address element do not cause extra white space to be inserted. (14)

The juxtaposition of these tags implies that the silence of the authors' part on the visual rendering of the hyperlink in this specification was not simply an omission.

Another hypothesis as to why there was no description of the preferred rendering of the hyperlink in the first proposed web standard is that the authors intended it to be platform-agnostic: many early users were accessing the web using non-graphical systems. However, that is belied by the inclusion of explicitly graphical content in other portions of the standard (although with the caveat, " Where not supported by implementations, like all tags, these tags should be ignored but the content rendered") (24). The renderings of the Heading tags (H1–H6) require the browsers to be able to justify text, present multiple font sizes, and print in bold and italic, albeit as suggested implementations (16). The specification also contains an entire section on character highlighting which define

the visual display of the given text in more or less explicit manner; sections include strong, italic, and underline.

Despite not providing an explicit rendering recommendation for hyperlinks, the first web standard proposal implicitly shaped the look of links. First, anchor tags are *tags*: they wrap a piece of text and provide additional semantic value to it, a property inherited from SGML. Any reasonable rendering in this standard would need to display not just that a link exists, but the extent of the text to which it is attached. A rendering such as putting an icon inline with the text would be semantically insufficient because it could only show where the link begins or ends, not what it encapsulates. For this reason, the specification negates an implementation such as what was used in NLS. If icons were used (as the specification suggests they could be) it would need to be in conjunction with another signifier. Second, and more generally, anchor tags are specific to a place in the hypertext. This property semantically precludes a list of hyperlinks at the end of the document, such as that used in Gopher or the NeXT help system. In these ways, the standard implicitly constrains the possible representations of hyperlinks that a web browser could implement while remaining semantically consistent with the intent of HTML.

The second web standard, HTML 2.0, was promulgated in November 1995 and continued to be mute on hyperlink styling. Interestingly, the underline tag (<U>) did not make it into this standard, and is listed as one possible browser extension ("Hypertext Markup Language - 2.0"). HTML 3.2 superceded 2.0 in early 1996; it included the underline tag, but continued to leave hyperlink representation up to browsers. It also makes provisions for the increasing role of style sheets in formatting screen output

("HTML 3.2 Reference Specification"). The HTML 4.0 standard, published in 1997, made a significant change in the ideology of visual representation of HTML with the relocation of visual attributes–such as color and fonts–from tags to style sheets defined using the Cascading Style Sheet (CSS) language ("HTML 4.0 Specification"). This iteration of the standard was also the first to give a suggestion on representing links visually. Section 12.1.3 dealing with anchors says that, "user agents may render this content in such a way as to indicate the presence of a link (e.g., by underlining the content)" which is the first time the notion that the underline could or should be used was mentioned. By this time, the web had grown substantially and web browsers were coalescing on best practices. Also, the visual browsers were by far the most common way to access the web, so accommodations for textual browsers were less central to the decision makers in W3C.

The current version of HTML, HTML5 was released in October 2014 and goes further than previous standards in trying to separate the semantic from presentational content of an HTML document. Berners-Lee, as the leader of the W3C organization, has made clear that his goal in improving the web is to make content more machine-readable and semantically oriented, and HTML5 clearly attempts to further that goal. However, the extensive installed base of web browsers and constellation of web servers also requires backwards compatibility to remain intact. HTML5 attempts to balance these two considerations and in many cases uses awkward retroactive continuity to elide the purpose of earlier visual elements. Deprecating these elements would rendering much web content technically invalid, so instead, they gave them a new life. For example, the HTML5 description of the <b> element, which the HTML 1.2 proposal describes

succinctly as “Boldface, where available, otherwise alternative mapping allowed, ” does not make any reference to bold at all, instead describing the element in section 4.5.18 as:

> a span of text to which attention is being drawn for utilitarian purposes without conveying any extra importance and with no implication of an alternate voice or mood, such as key words in a document abstract, product names in a review, actionable words in interactive text-driven software, or an article lede. (“HTML5”)

The section gives the advice that should be used where previously <b> would have been the proper choice, but includes the tag anyway as a compromise. The same pattern can be seen with the <u> tag. Defined in HTML 1.2 simply as “Underline,” in HTML5 its original purpose receives another lacuna. Its canonical meaning has morphed almost to the point of unrecognizability from its abbreviated root:

> The u element represents a span of text with an unarticulated, though explicitly rendered, non-textual annotation, such as labeling the text as being a proper name in Chinese text (a Chinese proper name mark), or labeling the text as being misspelt. (HTML5)

However, the original purpose is revealed in a cautionary note on how *not* to use the tag–a note which is also the first concrete reference in any HTML standard to the use of the underline for hyperlinks (HTML 4.0 only hinted at it). In section 4.5.19 about the <u> element, programmers are admonished:

> The default rendering of the u element in visual presentations clashes with the conventional rendering of hyperlinks (underlining). Authors are encouraged to avoid using the u element where it could be confused for a hyperlink. ("HTML5”).

The “pave the cowpaths” strategy which defined the early standards–many tags entered the standard as browser-specific tags which were canonized by W3C–is at once apparent in these changing usages and contradictory to them. Rather than abandoning the tag, <u> has absorbed new meaning. The standard also acknowledges the consensus, though not

officially endorsed, of the use of underlining to recognize hyperlinks. This standard essentially "gets out of the way" of the tide of common practice, while attempting to gracefully support vestigial uses of tags. This is an intelligent strategy, but very much in line with Berners-Lee's belief in making the web standards open, flexible, and decentralized.

The HTML standard has evolved significantly over the years, with an increasing emphasis on separating the semantic content of documents from their presentational aspects. In important ways, Berners-Lee's conception of HTML narrowed the range of possible hyperlink visualizations due to the SGML-like tag system. Throughout, the representation of hyperlinks to users has never been strictly prescribed: its implementation has always been left to the developer of browsers, or the creator of style sheets (on either the user- or publisher-side). However, the common use of underline to represent hyperlinks by developers became so pervasive that it forced W3C to revise its standards to reduce the clash of meanings. The discussion of web browsers that follows will show the important role that browsers had in leading to the coalescence on the underline.

The Demonstration Browsers

Tim Berners-Lee's specifications for the web, comprising of HTTP, URI, and HTML, would not have gained any traction without a reference implementation to demonstrate the web's capabilities. To this end, he built the first web browser, called WorldWideWeb in 1990. As he describes in *Weaving the Web***,** Berners-Lee hoped to find off-the-shelf software which could be modified to function with his newly defined web

protocols. Many hypertext editors existed, but they were primarily offline, single-user affairs. Berners-Lee looked at the Guide software by Owl Inc., which had many of the features he envisioned for a browser, but the company could not be convinced to edit the software to meet his specification (Berners-Lee and Fischetti 26). After striking out with multiple other software packages, he decided to write a new browser from scratch on the NeXT computer platform. The NeXT computer made rapid software development possible with its Interface Builder software and integrated networking support (discussed earlier in the chapter). Specifically, Berners-Lee took advantage of a quirk of the text editor widget of the Interface Builder: he used spare memory in the widget's representation of text to embed hyperlink data in the interface (28).

The WorldWideWeb browser[11] relied heavily on the interface widgets provided by NeXTSTEP, through the Interface Builder. Its menu bar consisted of a vertical stack of items, as was standard practice on NeXT, with options such as Document, Navigate, and Edit. The main window of the browser was based on the skeuomorphic metaphor of a blank page: Berners-Lee's vision was for the browser application to be equally fluent at reading and writing. Navigation (outside of following hyperlinks) was controlled through a navigation panel with options to travel back to the previous page, or use the previous/next functions to travel through a sequence of linked hypertext documents. The presentation of HTML tags was controlled by a style sheet packaged with the browser, although Berners-Lee envisioned users being able to modify their style sheet to suit their needs.

11. Appendix I describes the process of emulating NeXTSTEP on modern hardware and running the WorldWideWeb browser software.

CERN and Berners-Lee publicly released the source code for WorldWideWeb, and a close reading of the code yields interesting insight into the stylistic history of the browser. The default styling for an anchor is defined in the source code as being underlined. A commented-out line in the code of HyperText.m (the library which implements the display of hypertext within WorldWideWeb) shows that links were originally indicated by being a different color, slightly grey, but Berners-Lee deemed that solution "horrid" in a note in the source code and commented that style out ("HyperText Implementation"). The use of underline was easy to implement because these were capabilities of the Interface Builder's text editing widget. Furthermore, bold and italics were available as highlighting options for hypertext writers. The NeXT computer on which Berners-Lee developed has a grayscale video system, so no colors were used to highlight links–blue links were a later invention. This marks the very first usage of the underline to represent hyperlinks in a web browser, and thus on the web. The choice would prove to be momentous, as in short time all major browsers would follow suit.

However, the reach of WorldWideWeb was limited by the minimal installed base of NeXT computers; Berners-Lee realized that to increase the reach of the web, a text-based browser was needed. In 1990, a CERN intern programmed the "linemode" browser[12] to allow users of any computer system to access web servers. Since all contemporary computers were either text-based or at least could run a terminal window, no platform was left out (Berners-Lee and Fischetti 30). The web quickly spread at CERN due to the killer apps of a centralized phonebook and access to other databases[13]. Unlike the NeXT browser, the linemode browser eschewed any advanced styling, making

12. CERN maintains an emulated version of the linemode browser at info.cern.ch.

few assumptions about the capabilities of the terminal. When the browser came across formatting or styling instructions, it simply ignored them. Rather than underline, links were represented by bracketed numbers following an anchor tag (e.g. “Click here to more [1]”). This compromise was necessary to accommodate the text-only interface, but violates the principles of the HTML standard: the hyperlink signifier only indicates the end of the link span, introducing ambiguity and reducing the information-bearing content of the hypertext. However, time has proven that the web was sufficiently powerful and versatile that this browser “kludge” was not ruinous to the spread of the web, even though it was the first experience many had with the web.

Both the linemode browser and WorldWideWeb were intended to be demonstration browsers to help users understand the power of the web and spur development of production software. To expand the audience which could experience the web, at least in a cursory capacity, Berners-Lee set up a telnet server at CERN allowing users all over the world to access the web through the linemode browser without the hassle of installing it locally on their computers, or possessing the needed network software. From the beginning, Berners-Lee did not intend his software to be *the* portal to the web, but only a template to show what the web could do. Very quickly, other developers took the baton and ran with it: within a couple of years of the initial standard, the web was available with browsers on every platform. To foster the diaspora, Berners-Lee had his original Objective-C code rewritten in the more cross-compatible C language, which he considered to be a downgrade from the object-oriented native language of NeXT, but produced an easy to implement library which made the creation of web browsers on other platform simple (Berners-Lee and Fischetti 48). Developers

used this freely-released “libwww” library to port the functionality to a variety of other platforms which allowed the web to spread beyond CERN, thanks in great part to the organization's decision that it was not worth licensing (Gillies and Cailliau 210).

Early Browsers

A number of early browsers were developed quickly with the release of libwww, thanks to the coordination many individuals over internet message-boards. Information and excitement for the web spread rapidly, and developers could share their software with others over FTP easily. One of the first of this group of browsers, Erwise, was developed in Finland in 1992 and allowed the web to be accessed on Unix computers running the X Window system (Gillies and Cailliau 210). X Window was a graphical package available for Unix computers which allowed more advanced graphics than normal terminal-based output. While a relatively simple browser, it followed Berners-Lee’s pattern of using underlines to highlight links (“A Quick Look at Erwise”). MidasWWW, another X Window browser, added another now-obvious detail to the functioning of hyperlinks: the changing of color to represent that a link has already been visited (Gillies & Cailliau 225). A more successful browser was developed at UC Berkeley, also in 1992: violaWWW, originally an offline hypertext system called Viola, was modified to work on the web using for X Window by Pei Wei (Berners-Lee and Fischetti 56). The browser featured many extensions to HTML which were eventually adopted into the standard, such as the ability to display images inline with the text and an early applet system allowing execution of programs over the web (Gillies and Cailliau 214). Like the previous graphical browsers, violaWWW also used underlines to represent hyperlinks

(Wei). ViolaWWW made such an impression on Berners-Lee that he added a link to it to the original homepage for the web, info.cern.ch (Gillies and Cailliau 217). This imprimatur ensured that early Unix users with graphical capability would experience a version of the web which represented hyperlinks with an underline.

While the X Window browsers did bring the web to Unix systems, they were still limited to computers with graphical capabilities. This left users with text-based systems only with access to the sub-par linemode browser: interesting in concept, but not sufficiently enthralling to convert most people to web users. The Lynx browser, developed at the University of Kansas, changed that. Like violaWWW, it was originally developed as an offline hypertext navigator, but when it became obvious that libwww was becoming dominant, the developers adapted it to the web and released it as version 2.0 in 1993 (Berners-Lee and Fischetti 68). Lynx was text based, like the linemode browser, but rather than using numbered brackets to represent hyperlinks it underlined links. Lynx also has the option to use inverse video to highlight hyperlinks if that is what the system permitted (Gilster 161). Approximating the interface technique of the mouse, Lynx allowed use of the arrow keys to move the cursor around the screen with enter functioning as a click, activating the selected link (Gillies & Cailliau 232). Lynx took off on text-based Unix systems for the same reason the web had: the interface paradigm was simple and intuitive. Clicking directly on a highlighted piece of text makes the text itself the interface, which puts the user *into* the information space, and removes a level of mediation that earlier computer information systems imposed. The use of the underline balanced the subtlety of typography against the bulk of bolder interface devices such as

buttons or menus. The popularity of Lynx shows that the paradigm of the web was its driving force, rather than just the graphical flash of the NeXT WorldWideWeb browser.

While the web was born on a Unix based platform, it did not take long to be adapted to the other popular personal computing platforms. The first PC browser, Cello, was developed at Fermi National Accelerator Laboratory (Fermilab) in 1993 and was based off of libwww (Berners-Lee and Fischetti 69). Cello was an advanced browser which added new functionality including inline images and the ability to select the colors of elements on the page. In contrast to the other early browsers, Cello highlighted links with a dotted box around the selected text. Around the same time, the first Macintosh web browser, Samba, was released, also based on libwww. A very simple browser with minimal formatting capabilities, Samba represented links using boldface (Gillies & Cailliau 235). It is interesting that these two browsers–both trailblazers on their respective platforms–departed from the coalescing standard of highlighting hyperlinks on Unix systems. This outcome demonstrates that other hyperlinks modalities were possible, and tested, but ultimately were rejected when better software supplanted them.

Mosaic and its Spawn

The browser which brought the web to the masses (and almost supplanted it) was also pivotal in ossifying the use of the underline for hyperlinking. Released in 1993, NCSA Mosaic was developed at the National Center for Supercomputing Applications at the University of Illinois at Urbana Champaign, a government-funded computer research center (Berners-Lee and Fischetti 68). The first release of NCSA Mosaic operated in the X Window environment, as did many of its competitors. However, within a year of its

creation, a PC and Macintosh version were released and posted to the CERN website, still the de facto homepage of the web (Gillies & Cailliau 239). The browser was robust, full featured, and available across platforms, and it became popular almost immediately, to a great extent displacing the other graphical browsers. As Berners-Lee discussed in *Weaving the Web*, to his chagrin, so ubiquitous was NCSA Mosaic that it briefly became synonymous with the web in the public conversation. As Jakob Nielsen notes in 1995, "it is interesting to contemplate fact that Mosaic and the WWW more or less succeeded in establishing a universal hypertext system in just three years, even though Ted Nelson could not get his Xanadu system accepted in thirty years of trying" (Multimedia and Hypertext 1995). Mosaic's ubiquity, however, finally sealed its representation of the hyperlink–the underline[14]–as the standard.

The NCSA Mosaic browser was not only the graphical inspiration for the next generation of web browsers, but also their actual textual source. The Mosaic group spun off from NCSA to create an even better browser in 1994 (Gillies & Cailliau 256). The resulting product, Mosaic Netscape (eventually just Netscape) was based on the NCSA Mosaic code and thus resembled it very closely, including its link presentation. The rapid adoption of new browsers is captured in an early longitudinal web survey conducted by the GVU Center at the College of Computing of Georgia Institute of Technology. The January 1994 version of the survey found that almost all of the respondents used Mosaic to access the web (although their survey methodology was biased towards Mosaic-using respondents). By October 1994, 18% of respondents were using Netscape, and by the

14. Some screenshots available on the web show very early versions of the PC version of Mosaic using blue-colored text without an underline as the hyperlink signifier. This appears to have been changed to match the other platform version early on.

April 1996 survey, almost 90% of users “expected to use Netscape in the next 12 months.” In just a couple of years, the web’s browser landscape was upturned.

Internet Explorer had an equally rapid domination of the browser market, starting with the same genes as Netscape. Microsoft, seeing the growing popularity of the web, wanted to the enter the market rapidly and to do so intended to include a web browser with its upcoming Windows 95 software (Berners-Lee and Fischetti 108). Rather than writing its own software from scratch, the company elected to licence the core code of the browser from Spyglass, Inc. which itself licensed it from NCSA (Gillies & Cailliau 258). The major browsers at this time were not just competing for market share, they were competing for pieces of the rapidly increasing pie, but both major browsers had descended from the same ancestor. By 1996, the GVU study reported that almost 60% of users expected to use Internet Explorer within the next year. Internet Explorer also used the underline to represent hyperlinks.

Without the benefit of retrospect, the contemporary press seemed very confused about what the web actually was. The conflation of web with Mosaic is just one example. In his book *Finding it on the Internet*, Paul Gilster writes about the web from the perspective of line browsers primarily, with graphical browsers being an outgrowth from them. He appears to be mostly familiar with using earlier systems like Gopher, and is unaware of the inherently graphical origins of the web. When he mentions Mosaic, he seems to demean it as a superficial extension to the web, which in his view is, at its core, text oriented. For example, "the difference between Mosaic and a standard, character-based browser like www..." (169). With the popularity of the linemode browser and then Lynx, many early web users accessed the web through textual means and did not realize

it had graphical, mouse-based origins. Of course, NeXT adoption was low, and the Macintosh had not yet established extensive market share. Ironically, though, Gilster acknowledges that formatting content is "lost" when browsing with VT-100, but he still presents that as primary and the graphic browser as additional.

The progression of web browsers has continued with multiple generations of browser technology replacing the next, each one intertwined with the previous, both visually and through direct code transfer. That history has been documented and continues to evolve, but the die was cast in the earliest days of web browsers. By and large, browsers since Netscape and Internet Explorer have continued to use underlines to represent hyperlinks, despite alternative options occasionally being tested.

Usability and the Future

The early years of academic interest in hypertext preceded the invention of the web, but that interest was ultimately overshadowed by its dominance as the premier global hypertext system. Academic research investigated many aspects of the user interface to hypertext systems, including the optimal method of highlighting links. Evenson, Rheinfrank, and Wulff addressed this problem as early as the 1989 Hypertext Conference in their article “Towards a Design Language for Representing Hypermedia Cues.” Their analysis enumerated many different possible ways to represent hyperlinks, using as a jumping off point their observation that “most systems have used bold, italic or underlined text to indicate a word is that is hyper...” but worrying that “these typestyles in conjunction with plain text would disrupt the flow of text on the page (or screen) and interfere with readability” (86). To solve this problem they proposed a variety of possible

solutions, including a new type of textual highlighting, before proposing “a new graphic sign-set for the representation of hypermedia cues...based on the set of potential actions within hypermedia systems” (89). Their set included an icon for each possible connection between two nodes in a hypertext that they could think of. A sketch of their proposed system shows a textual hyperlink surrounded by a box, with one of their proposed icons interposed within it. The authors viewed their system as necessary because they envisioned hypertext systems flourishing with a robust set of possible meanings attached to a hyperlink. Ultimately, the simplicity of the web negated the need for multiple user-visible hyperlink signifiers: a link was monodirectional and monotypical.

Patricia Wright also approached this question on an analytical level in her article “Interface Alternatives for Hypertext,” published in the journal *Hypermedia* in 1989. Her approach was also birthed in an environment where the default assumption was that a hypertext system would have a complex link modality. Wright attempts to categorize all of the dimensions of hypertext interfaces along a “reader-limited→reader-free” axis, looking at factors such as link display, link overview, inline versus menu-based links, and hyperlink type. Like the Evenson, et al. article, Wright considered the selection of link visualization a very open question, and non-obvious. Both of these articles assumed that the user would be “lost in (hyper)space” and that the hypertext creator would need to develop extra tools and demarcations to give context to each link and page within hyperspace. The concept of an overview map of the entire space was also floated in many early discussions of hypertexts. Nielson explains how the hyperlink was sufficient for its purpose on the web, and that users perform faster when links are in hypertext rather than menu:

> hypertext functions as "embedded menu". . . [where]...parts of the primary text or graphics does double duty as being both information in itself and being the link anchor. (Multimedia and Hypertext 138)

The simplicity of the web proved that these fears were unfounded, and a single identifier for the hyperlink was sufficient. However, even with a single hyperlink signifier, there was concern that choosing the wrong one could conflict with traditional typographical usage. Evenson, et al. proposed a whole new typographic style, while Nielson presciently acknowledged that:

> Unfortunately the highlighting of anchors conflicts with the use of emphasis in the running text. Traditional writers have used typographical notations like italics or boldface type to indicate various forms for emphasis or special purpose text like quotations…[b]ut many current hypertext systems use the same or similar notation to indicate hypertext anchors also…One solution to this problem may be …the gradual emergence of conventions for hypertext notation. (Multimedia and Hypertext 140)

The simple solution won out over a more complex navigation system due to its ease of use and, both for content creators and readers. Nielson's prognostication was not crystal clear, however, as he predicted that websites would need to be highly hierarchical to be navigable, although he was writing before powerful search engines became available (Multimedia and Hypertext 256). By 2004, Nielson's first prediction has come to fruition and using underlining for links has become best practice, per his popular usability website Nielsen Norman Group ("Guidelines for Visualizing Links").

While the question was *de facto* settled on the web, academia continued to analyze the optimal way to display hyperlinks. In 2001, in addressing W3's XLink proposal (for XML), Weinreich, et al. set to investigate whether it would be possible to standardize link display. The paper enumerates the various colors, icons, outlines, fonts, etc. used in earlier browsers and attempts to envision ways that a more rich linking

system could be displayed–harking back to the hypertext research from the late 1980s and early 1990s. In 2010, Tullis and Siegel penned a paper “Does Underlining Links Help or Hurt?” looking at whether the underline was still a good piece of the web’s user interface. The result of the experimental study was mixed, with the results showing that in some cases links aid usability and in others they do not, although the effect was small.

The design of the web is prone to trends, but is always mediated by its technological underpinnings. The rise of the mobile browsing has changed the way that much of web content is consumed. The smaller screen sizes and the use of touch screen instead of a mouse have altered the affordances of the web in ways that are still being studied. One could envision a schism between interface modes on mobile versus desktop browsers, perhaps the extra pixels to display and underline are better spent elsewhere on mobile. Or, users will become so used to hypertext that they can suffice with subtle clues only. Time will show whether the underline is or is not a long standing feature of the web into the mobile era, but designers are sure to continue innovating.

Chapter IV

Conclusion

As the first chapter demonstrated, the underline has a long history in human writing and has always acted as a *vinculum* tying text together into semantically meaningful units. Early computer systems adopted the use of the underline for various purposes primarily because it was technically simpler to implement than alternative modes of highlighting, and also because it was an easily understandable user interface element. Early hypertext systems before the web used a variety of modalities to represent links depending on the hardware available to the developers, these modes included underlining, bold, reverse video, boxes, color, and many more. Tim Berners-Lee's development of the web vastly expanded the reach of hypertext systems, and thanks to the hardware affordances of the Interface Builder software packaged with the NeXT computer, on which the first web browser was built, the underline was used to show links. This initial condition set into motion a cascade of other web browsers modelled, more or less directly, off of Berners-Lee's. While there was some variability in their link representation, within a couple of years a steady state was reached with a progression of hegemonic browsers all adopting the underline (Mosaic, Netscape, Internet Explorer), despite the representation never being formally standardized. The actual code of the early browsers (libwww) was directly used in other browsers, encouraging developers to hew even closer to Berners-Lee's vision. The basic visual representation of the web was even ported successfully to non-graphical systems, most successfully in the form of Lynx.

Tim Berners-Lee's design of HTML as well as his demonstration browser also shaped the visual look of the web, although to a lesser extent. The use of SGML as the

markup language for HTML defined links as a *span of content* rather than a *point within content*. This structure encouraged an inline representation of links, rather than a menu/list system, or one interspersing text with icons or buttons: both modes implemented by other contemporary browsers. Berners-Lee's HTML specification was intentionally silent on the visual representation on hyperlinks, and did include multiple link types to represent semantic information about monodirectional links, although these were not implemented in a generally useful way in early browsers.

At no point in the development of the web was a large, thorough test of all hyperlink visualization modalities conducted. Instead, the pragmatic need to implement his vision as a usable product motivated Berners-Lee to use the best tools available to him to create a browser with preexisting tools. The technical affordances of the available hardware constrained and shaped the software and eventual medium created on it. This observation supports the thesis of George Basalla's *Evolution of Technology*, which posits that most technologies, even revolutionary ones, are built upon and shaped by earlier technical innovations. The "heroic" inventor, while vital to the creation of the given invention of technology, was working within a specific milieu and with access to certain fundamental existing technology. In the case of Tim Berners-Lee, he was aware of existing hypertext projects and CERN provided him with a highly networked environment in which to build and test his concept. The NeXT computer on which he developed his browser made the use of underlining simple, and its existing semantic meaning of overloading its referent with additional content made it logical and sufficient to the task of representing a hypertext span.

This study intends to add to two related areas of study. First, it has explored the semiotic meaning of the underline through a survey of its use through history and across disciplines, including mathematics, commerce, printing, and user interface design. Secondly, this study has shown how the underline was adopted to be the global hyperlink signifier, expanding, but not fundamentally changing the meaning of the underline in textual contexts, despite facing challengers from other symbolic representations. Future studies may be warranted to investigate the use of the underline in mobile contexts, or future implementations of web browsers beyond the traditional confines of a desktop browser. At the least, I hope it adds a piece to the stories of each topic.

Appendix

NeXT Emulation Process

In order to examine the WorldWideWeb browser in as close to its "native habitat," as possible, but with NeXT computer in short supply, I used emulation software to run the operating system on modern hardware. The main challenge with emulation NeXT hardware is that is used the proprietary Motorola 68000 series of processors. Most modern desktop computers use x86 processors which use a totally different hardware instruction set. Code written for NeXTStep on the 68000 processor is not compatible with modern computers, and would need to be recompiled to work. Later versions of NeXTStep and its successor OpenStep were ported to x86, but existing software (for instance WorldWideWeb) would not run. Running one of the ported version of NeXTStep /OpenStep is not very challenging and can be achieved using virtualization software such as VMWare or VirtualBox. I initially installed NeXTSTEP 3.3 on VirtualBox to explore the environment, but the CPU instruction set incompatibility made this futile. To run the original software it is necessary to use emulation software which simulates the 68000 hardware on a modern computer.

There exists a very active group of NeXTStep enthusiasts who have attempted to keep the system alive and are centered around the web forum, nextcomputers.org. This group has created an emulator called Previous (a play on words with NeXT) which successfully emulates the original Next hardware including peripherals. I used this

software along with an disc image of the original NeXTSTEP version 3.3 operating system installation CD-ROM (acquired from Benjamin Woodley's website).

The following resources were used to install a functioning NeXTStep installation:

Instructions.

- "NEXTSTEP 3.3 install in VMware on OSX part-1." youtu.be/dj8Hyz-7arU
- "Ben's VirtualBox Help Page." by Benjamin Woodley
- "Installing and running NextStep/OpenStep (Intel) on PC emulators" by Tomaž Slivnik onnextcomputers.org/

Software.

- Browsers: nextcomputers.org/NeXTfiles/Software/
- Boot disks: nextcomputers.org/NeXTfiles/Software/NEXTSTEP/Floppy_Images/
- Installation CD-ROM ISO: sites.google.com/site/benvirtualbox/home/nextstep

Bibliography

Abadie, Michel. *Minitel Story: Les Dessous D'un Succès-*. Lausanne, Suisse: P.-M. Favre, 1988. Print.

Adler, Michael H. *The Writing Machine*. London: Allen & Unwin, 1973. Print.

Aker, Sharon Zardetto, Arthur Naiman, and Dale Coleman. *The Macintosh Bible: Thousands of Basic and Advanced Tips, Tricks, and Shortcuts, Logically Organized and Fully Indexed*. 3rd ed. Berkeley, Calif.: Goldstein & Blair, 1991. Print.

Bardini, Thierry. *Bootstrapping: Douglas Engelbart, Coevolution, and the Origins of Personal Computing*. Stanford, Calif.: Stanford UP, 2000. Print. Writing Science.

Barnet, Belinda. "Crafting the User-Centered Document Interface: The Hypertext Editing System (HES) and the File Retrieval and Editing System (FRESS)." *Digital Humanities Quarterly* 4.1 (2010). Alliance of Digital Humanities Organizations. Web.

Basalla, George, ed. *The Evolution of Technology*. Cambridge, UK: Cambridge University Press, 1988.

Baudrillard, Jean. *Simulacra and Simulation*. Ann Arbor: U of Michigan, 1994. Print. The Body, in Theory.

Berners-Lee, Tim. “HyperText Implementation.” *World Wide Web Consortium (W3C)*. May 1991. Web. 23 May 2016.

---. “A Quick “Look at Erwise.” *World Wide Web Consortium (W3C)*. Web. 23 May 2016.

Berners-Lee, Tim, and Daniel Connolly. "Hypertext Markup Language (HTML): A Representation of Textual Information and MetaInformation for Retrieval and Interchange." *World Wide Web Consortium (W3C)*. June 1993. Web. 22 May 2016.

Berners-Lee, Tim, and Daniel Connolly. “Hypertext Markup Language - 2.0.” *World Wide Web Consortium (W3C).* November 1995. Web. 22 May 2016.

Berners-Lee, Tim, and Mark Fischetti. *Weaving the Web: The Original Design and Ultimate Destiny of the World Wide Web by Its Inventor*. San Francisco: HarperSanFrancisco, 1999. Print.

Berners-Lee, Tim, et. al. "World-Wide Web: An Information Infrastructure for High-Energy Physics." Proceedings of the Second Workshop on Artificial Intelligence and Software Engineering for High Energy Physics, La Londe, France, January 1992. New Computing Techniques in Physics Research, World Scientific, Singapore. pp157-164.

Birks, Peter, and Eric Descheemaeker. *The Roman Law of Obligations*. First ed. 2014. Print. Collected Papers of Peter Birks.

Bush, Vannevar. “As We May Think.” *The Atlantic* July 1945. *theatlantic.com* Web. 6 Mar. 2016.

Card, Stuart K., Allen Newell, and Thomas P Moran. *The Psychology of Human-computer Interaction*. Hillsdale, N.J.: L. Erlbaum Associates, 1983. Print.

Carter, Harry, and James Mosley. *A View of Early Typography up to about 1600*. Hyphen, 2002.

Cats-Baril, William, and Tawfik Jelassi. "The French Videotex System Minitel: A Successful Implementation of a National Information Technology Infrastructure." *MIS Quarterly* 18.1 (1994): 1. Web.

Connolly, Edward S., and Lieberman, Philip. *Introducing the Apple Macintosh*. 1st ed. Indianapolis, Ind.: H.W. Sams, 1984. Print.

Current, Richard Nelson. *The Typewriter and the Men Who Made It*. Urbana: U of Illinois, 1954. Print.

Dawkins, Richard. *The Selfish Gene*. New ed. New York: Oxford UP, 1989. Print.

Dewey, Melvil. *Library School Rules*. Boston: Library Bureau, 1890.

Dodd, Robin. *From Gutenberg to Opentype: An Illustrated History of Type from the Earliest Letterforms to the Latest Digital Fonts*. London: Ilex, 2006.

ECMA: European Computer Manufacturers Association. "Standard ECMA-48: Additional Control Functions for Character-Imagine I/O Devices." ECMA International, August 1971. Web. 20 Apr. 2016.

Engelbart, Douglas C.. *A Research for Augmenting Human Intellect*. Stanford Research Institute, 1968. Online. *MouseSite*. Stanford University. Web. 22 Mar. 2016.

Epstein, Ralph C. "Industrial Invention: Heroic, or Systematic?" *The Quarterly Journal of Economics* 40.2 (1926): 232-72. Web.

Evenson, Shelley, John Rheinfrank, and Wendie Wulff. (1989). “Towards a Design Language for Representing Hypermedia Cues.” In: Halasz, Frank, Meyrowitz, Norman (eds.) Proceedings of ACM Hypertext 89 Conference November 5-8, 1989, Pittsburgh, Pennsylvania. pp. 83-92.

Fischer, Eric. “The Evolution of Character Codes, 1874-1968.” Web.

Garfinkel, Simson, and Michael K. Mahoney. *NeXTSTEP programming: Step One, Object-Oriented Applications*. New York: Telos Springer-Verlag, 1993. Web. http://simson.net.

Gillies, James, and R. Cailliau. *How the Web Was Born: The Story of the World Wide Web*. Oxford: Oxford UP, 2000. Print.

Gilster, Paul. *Finding It on the Internet: The Internet Navigator's Guide to Search Tools and Techniques*. Rev. and Expanded, 2nd ed. New York: Wiley, 1996. Print.

Gladwell, Malcolm. “Creation Myth.” *The New Yorker* May 16, 2011. newyorker.com. 22 Mar. 2016.

“GP IBM Selectric Typeball Catalog.” monk.org. March 1986. Web. 7 April 2016.

"GVU's WWW User Surveys." *GVU Center's WWW User Surveys.* GVU Center, College of Computing Georgia Institute of Technology, 29 June 2001. Web. 31 May 2016.

Hart, Glenn A. "IBM Sets a New Standard." *PC Magazine* 25 Dec. 1984: 172-78. *Google Books*. Web. 22 Mar. 2016.

Hiltzik, Michael A. *Dealers of Lightning: Xerox PARC and the Dawn of the Computer Age*. 1st ed. New York: HarperBusiness, 1999. Print.

Hindman, Sandra, and James Douglas Farquhar. *Pen to Press: Illustrated Manuscripts and Printed Books in the First Century of Printing*. Print.

Hirsch, Rudolf. *Printing, selling and reading, 1450-1550*. Otto Harrassowitz, 1974. Print.

Holzman, Daniel. “The Selectric Connection.” *PC Mag,* vol 3, num 3 (1984). Accessed via Google Books. 21 October 2016.

Hopkins, Judith. "The 1791 French Cataloging Code and the Origins of the Card Catalog." Libraries & Culture 27.4 (1992): 378-404. Web. 28 February 2016.

Houston, Keith. *Shady Characters: The Secret Life of Punctuation, Symbols, & Other Typographical Marks*. First ed. 2013. Print.

"HTML 3.2 Reference Specification." HTML 3.2 Reference Specification. Ed. Dave Raggett. World Wide Web Consortium (W3C), 14 Jan. 1997. Web. 22 May 2016.

"HTML 4.0 Specification." HTML 4.0 Specification. Ed. Dave Raggett, Arnaud Le Hors, and Ian Jacobs. World Wide Web Consortium (W3C), 18 Dec. 1997. Web. 22 May 2016.

"HTML5." HTML5. Ed. Ian Hickson, et al. World Wide Web Consortium (W3C), 28 Oct. 2014. Web. 22 May 2016.

"IBM Typewriter Milestones." *IBM Archives*. IBM, n.d. Web. 18 July 2016.

Ifrah, Georges. *The Universal History of Numbers: From Prehistory to the Invention of the Computer*. New York: J. Wiley, 2000. Print.

International Business Machines Corporation. *IBM Personal Computer Hardware Reference Library Technical Reference*. 1983.

International Business Machines Corporation (2). *IBM Personal Computer Hardware Reference Library IBM Color/Graphics Monitor Adapter*.

International Business Machines Corporation (3). *IBM Personal Computer Hardware Reference Library IBM Color/Graphics Monitor Adapter*. 1991.

"italic". *Encyclopædia Britannica. Encyclopædia Britannica Online.* Encyclopædia Britannica Inc., 2016. Web. 21 Mar. 2016.

Keppie, L. J. F. *Understanding Roman Inscriptions*. Baltimore: Johns Hopkins UP, 1991. Print.

Kessler, Jack. "The French Minitel: Is There Digital Life Outside of the "US ASCII" Internet? A Challenge or Convergence?" *D-Lib Magazine* 1.6 (1995): D-Lib Magazine, 12/1995, Vol.1(6). Web.

Krajewski, Markus. *Paper Machines: About Cards & Catalogs, 1548-1929*. Cambridge, Mass.: MIT, 2011. Print. History and Foundations of Information Science.

Loison, Fabien. "Bash Tips: Colors and Formatting (ANSI/VT100 Control Sequences)." *FLOZz*. Web. 21 Apr. 2016.

March, H. Colley. "The Meaning of Ornament; or Its Archaeology and Its Psychology." *Transactions of the Lancashire and Cheshire Antiquarian Society* 7 (1890): 160-92. *Google Books*. Web. 15 Dec. 2015.

Microsoft. "Code Page 437 MS-DOS Latin US" Microsoft Developers Network. Web. 21 Apr. 2016.

Nicolson, C. L., and C. J Bowden. New Zealand. Applied Mathematics Division. *I.B.M. Selectric Typewriter Interface System*. Wellington, N.Z.: Applied Mathematics Division, Dept. of Scientific and Industrial Research, 1976. Technical Report (New Zealand. Applied Mathematics Division) ; No. 48.

Nielsen, Jakob. *Hypertext and Hypermedia*. Boston: Academic, 1990. Print.

---. *Multimedia and Hypertext: The Internet and beyond*. Boston: AP Professional, 1995. Print.

---. "Guidelines for Visualizing Links." Nielsen Norman Group, 10 May 2004. Web. 01 June 2016.

Nielsen, Jakob, and Kara Pernice. *Eyetracking Web Usability*. Berkeley, Calif.: New Riders, 2010. Web.

Obendorf, Hartmut and Harald Weinreich. "Comparing Link Marker Visualization Techniques: Changes in Reading Behavior." *World Wide Web Proceedings of the 12th International Conference* (2003): 736-45. Web.

Ogburn, W., and Dorothy Thomas. "Are Inventions Inevitable?" *Political Science Quarterly* 37 (1922): 83. Print.

Pinch, Trevor J., and Wiebe E. Bijker. "The Social Construction of Facts and Artefacts: Or How the Sociology of Science and the Sociology of Technology Might Benefit Each Other." *Social Studies of Science* 14.3 (1984): 399-441. Print.

Pugh, Emerson W., Lyle R Johnson, and John H Palmer. *IBM's 360 and Early 370 Systems*. Cambridge, Mass.: MIT, 1991. Print. History of Computing.

Sayare, Scott. "On the Farms of France, the Death of a Pixelated Workhorse." *The New York Times.* 28 June 2012: A8. Print.

"Semantics, Structure, and APIs of HTML Documents." *HTML Living Standard*. WHATWG, 20 Dec. 2015. Web. 20 Dec. 2015.

Smith, David Eugene. *History of Mathematics*, *Volume 2*. Dover Publishing, Inc. New York. 1958. Print.

Stross, Randall E. *Steve Jobs and the NeXT Big Thing*. New York: Atheneum; Maxwell Macmillan International, 1993. Print.

The Chicago Manual of Style. 15th ed. Chicago: U of Chicago, 2003. Web.

The Chicago Manual of Style. 16th ed. Chicago: U of Chicago, 2010. Web.

"The Unicode Standard Version 8.0 – Core Specification." *The Unicode Consortium*. Ed. Julie D. Allen. Aug. 2015. Web. 20 Apr. 2016.

Tufte, Edward R. *The Visual Display of Quantitative Information*. 2nd ed. Cheshire, Conn.: Graphics, 2001. Print.

Tullis, Tom, and Marisa Siegel. "Does Underlining Links Help or Hurt?." *CHI'10 Extended Abstracts on Human Factors in Computing Systems*. ACM, 2010.

University of Chicago. Press. *Manual of Style, Being a Compilation of the Typographical Rules in Force at the University of Chicago Press, to Which Are Appended Specimens of Types in Use.* Chicago: U of Chicago, 1906. Print.

University of Chicago. Press. *Manual of Style, a Compilation of the Typographical Rules in Force at the University of Chicago Press, with Specimens of Types in Use.* Chicago: U of Chicago, 1910. Web.

University of Chicago. Press. *Manual of Style; a Compilation of Typographical Rules Governing the Publications of the University of Chicago, with Specimens of Types Used at the University Press.* 5th ed. Chicago, Ill.: U of Chicago, 1917. Web.

University of Chicago. Press. *A Manual of Style: Containing Typographical Rules Governing the Publications of the University of Chicago, Together with Specimens of Type Used at the University of Chicago Press.*10th ed. Chicago, Ill.: U of Chicago, 1937. Print.

University of Chicago. Press. *A Manual of Style Containing Typographical and Other Rules for Authors, Printers, and Publishers Recommended by the University of Chicago Press, Together with Specimens of Type.* 11th Ed., [rev.]. ed. Chicago: U of Chicago, 1949. Print.

University of Chicago. Press. *A Manual of Style: For Authors, Editors and Copywriters.* 12th Ed., Rev. ed. Chicago: U of Chicago, 1969. Print.

University of Chicago. Press. *The Chicago Manual of Style: For Authors, Editors, and Copywriters.* 13th Ed., Rev. and Expanded. ed. Chicago: U of Chicago, 1982. Print.

Vaughan, Tay. *Using HyperCard: From Home to HyperTalk*. Carmel, Ind.: Que, 1988. Print.

Webster, Bruce F. *The NeXT Book*. Reading, Mass.: Addison-Wesley, 1989. Print.

Wei, Pei-Yuan. “Viola Web Browser.” Web. 24 May 2016.

Weinberger, David. *Everything is Miscellaneous: The Power of the New Digital Disorder*. New York: Times, 2007. Print.

Weinreich, Harald, Hartmut Obendorf, and Winfried Lamersdorf. "The Look of the Link - Concepts for the User Interface of Extended Hyperlinks." *Hypertext and Hypermedia Proceedings of the 12th ACM Conference* (2001): 19-28. Web.

Wershler-Henry, Darren S. *The Iron Whim: A Fragmented History of Typewriting*. Toronto: McClelland & Stewart, 2005. Print.

Wiegand, Wayne A. *Irrepressible Reformer*: A Biography of Melvil Dewey. Chicago: American Library Association, 1996. Print.

Wright, Alex. *Cataloging the World: Paul Otlet and the Birth of the Information Age*. 2014. Print.

Wright, Patricia. "Interface Alternatives for Hypertexts." *Hypermedia* 1.2 (1989): 146-66. Web.