THE FORM OF THE WEB BROWSER AND ITS SOCIAL EFFECTS

A Thesis
submitted to the Faculty of the
Graduate School of Arts and Sciences
of Georgetown University
in partial fulfillment of the requirements for the
degree of
Master of Arts
in Communication, Culture, and Technology

By

Matthew T. Marco, B.A.

Washington, DC
April 15, 2011

THE FORM OF THE WEB BROWSER AND ITS SOCIAL EFFECTS

Matthew T. Marco, B.A.

Thesis Advisor: David Ribes, Ph.D.

ABSTRACT

While academics and journalists discuss how particular web applications (e.g. Google, YouTube, and Facebook) and hardware forms (e.g. smartphones) affect culture and behavior in networked societies, they pay little attention to the role browsers play as a common platform and interface for these applications and hardware. When we surf the web, we use browsers, and this thesis' two case studies will provide a starting point for analysis of how the form of the browser, expressed in its interface and underlying technologies, structures people's experiences of the web. By studying browsers as technological artifacts, this thesis will reveal how the experience of the web is structured by browser makers. Through the social construction of technology (SCOT) theoretical framework, I will broaden the definition of browser makers to include executives at technology startups and established firms, standards writers, graphic designers, and government regulators in addition to software architects and engineers.

Drawing from lectures, interviews, email archives, court proceedings, and journalism from the browser wars, I will describe how two technical achievements of the early modern web – Netscape's implementation of HTML's table element and the invention of JavaScript – gave rise to uses of the web as a design medium and a computing platform. Browser programmers' proprietary modifications to HTML perpetuated the metaphor of the page by encouraging documents on the web to be treated as two-dimensional canvases. By adding support for images

and grids in HTML, browser programmers allowed graphic design practices to exist on the web and accelerated its adoption for commercial uses. This advance preceded actions by the World Wide Web Consortium (W3C) and Internet Engineering Task Force (IETF) on the development of graphic design for the web, created de facto standards for web design, and minimized the authority of the W3C and IETF in shaping the web. The idea of the browser as a computing platform followed Netscape's development of JavaScript, a vivid example of how a market strategy to compete against Microsoft's "natural" monopoly in operating systems motivated the design of what would become the world's most popular programming language and the foundation for interactive applications on the web. While JavaScript is now a standard and an essential part of the web, its formalization as Ecma standard E-262 is a case of a *de facto* standard being submitted for adoption by a standard-setting body in order to create a competitive advantage.

These are just two of numerous conventions that came to constitute the form of the browser, and web browsers either adopt or respond to these standards. Through analysis of how these features and languages became conventions and web standards, this thesis provides a platform and interface for discussions of how the web affects the people who use it.

A NOTE ON STYLE

Grammar and punctuation may not be identical between quoted material and original writing. When I write the words “internet” and “web,” they shall be lowercase. “World Wide Web” as it refers to an application of the internet shall be treated as a proper noun. When these words are used in quotations, their original cases shall be preserved to denote the style appropriate at the time when it was written.

“Link” is equivalent to “hyperlink,” though original references in quotations will be preserved. “Browser” and “web browser” are also identical. “Site” and “website” are also identical, though in this case, I intend to use the latter. Note that “website” is written here as one compound word and treated as a common noun.

This style is influenced by *Wired* magazine’s house style in place since 2004. *Wired* copy chief Tony Long wrote when announcing the change:

> In the case of internet, web and net, a change in our house style was necessary to put into perspective what the internet is: another medium for delivering and receiving information. That it transformed human communication is beyond dispute. But no more so than moveable type did in its day. Or the radio. Or television.
>
> This should not be interpreted as some kind of symbolic demotion. Think of it more as a stylistic reality check.

A NOTE ON HISTORY

These are not strict definitions of epochs, but rather, ways to quickly refer to different kinds of work done by browser makers at different times in the history of the web. A history of the period covered by this thesis can be found in *Competing on Internet Time* by Michael Cusumano and David Yoffie, Appendix One.

- Early hypertext: historical examples such as the memex, NLS, and Xanadu.
- Pre-modern browser: 1991-1994.
- Modern browser: 1995-present
    - Early modern: 1995-2002 (browser wars)
    - Standards-era: 2003-2008 (web standards, Web 2.0)
    - Contemporary: 2009-present (HTML5, CSS3, web fonts, video, audio, ubiquity, ‘death of the web’)

For my friends and colleagues, especially Christina Lee.

Table of Contents

INTRODUCTION

> [T]he Web is now 18 years old. It has reached adulthood. An entire generation has grown up in front of a browser. The exploration of a new world has turned into business as usual. We get the Web. It's part of our life.
>
> –Chris Anderson, editor-in-chief, Wired, "The Web Is Dead. Long Live the Internet." September 2010
>
> "The irony is that in all its various guises - commerce, research, and surfing - the Web is already so much a part of our lives that familiarity has clouded our perception of the Web itself. To understand the Web in its broadest and deepest sense, to fully partake of the vision that I and my colleagues share, one must understand how the Web came to be."
>
> –Tim Berners-Lee, inventor of the World Wide Web, Weaving The Web, 1999

It is no exaggeration to say that in the two decades since it was invented, the World Wide Web has become part of everyday life. People communicate with peers on social networking sites, publish through blogging services, shop, apply for mortgages, and even file their annual tax returns on the web. Despotic governments either cut off or tightly control internet access to exert control over their citizens.[1]

Subsequent to the web's uses in daily life, implementations of standards affect the lifeblood of enormous businesses and the flow of vast sums of money and capital. Netscape Communications Corporation, who shipped the first modern web browser, Navigator, as a public beta in 1994, set a first-day growth record for an IPO[2] of its size (and a company with no record of profit). Four years later, the company was acquired by America OnLine for $4 billion. It was only the first of numerous public offerings by web-based startups with similar balance sheets – in 1999 alone, 308 startups went public, and the largest 24 of them had a combined market capitalization of $70 billion.[3] In March 2011, five of the largest web-based companies yet to go public had a total valuation of $71.3 billion.[4] Google – founded around a web search engine – boasts a market value of $148 billion.[5] Amazon – founded as a web-based bookstore and now among the world's largest retailers – generated nearly $13 billion in revenue in 4Q 2010.[6] EBay restated their quarterly earnings forecast following a 2004 redesign of their new user registration form because the improvement of the web-based process had generated such a dramatic increase in sales.[7] Implementations of web standards – *de jure* technical specifications for the web's bedrock programming languages – affect these companies daily operations and new specifications create new territory for established firms or startups to claim.

The growth of businesses founded on the web is mirrored in the unknown (and growing) total breadth of the web. The number of pages on the web that search engines have indexed exceeds one trillion; this does not include the contents of the various databases that websites use to tailor content for their users.[8] Despite this breadth, the web's incomprehensibly large collection of pages marked-up and gathered into sites and interactive applications are all accessed through a common application, the browser.

This thesis will provide a starting point for analysis of how the form of the browser, expressed in its interface and underlying technologies, structures people's experiences of the web. As firms and individuals have sought to change the web to address particular problems, they have developed solutions through the browser. Google, which maintains a suite of cloud-based web applications, has sought to advance the state of the web through their entrance in the browser market with the development and release of their own browser, Chrome, in 2008.[9] For Apple, their decision to not support Adobe Flash – a third-party "plug-in" to enhance browser functionality – in the mobile version of their Safari browser released in 2007 solved a more precise "chicken-and-egg problem" related to the distribution of videos and interactive applications:

> Publishers use Flash for web video because Flash is installed on such a high percentage of clients; clients support Flash because so many publishers use Flash for web video. Apple … is solving the chicken and egg problem. For the first time ever, there is a large and growing audience of demographically desirable users who don't have Flash installed. If you want to show video to iPhone users, you need to use H.264.[10]

The developments of Chrome and Safari echoed Netscape's impetus for implementing new elements in HTML – the building blocks of every page on the web – and developing the JavaScript programming language in the mid 1990s. Through the process of making browsers, executives and programmers at Netscape made decisions about how to implement HTML not just by giving a voice to its vocabulary of elements but by making new elements and even new languages to address users' concerns.

Browser makers implement a vision for the web that interprets and sometimes disregards existing standards and, in the products they ship, define the boundaries within which users construct the web. Through the social construction of technology (SCOT) theoretical framework, this thesis broadens the definition of *browser makers*. In addition to those companies who shipped browsers as software products, this thesis will reveal how various social groups constructed the tools by which people communicate and create on the web. The vocabulary of HTML and JavaScript and the interface of the browser impose boundaries on the forms of pages and the structure through which users navigate them.

The browser, in its consistent role as (literally) the frame of the web, offers an opportunity to glimpse the evolution of the web through the construction of the technological artifact that mediates it. It is the interface that mediates our usage of the web, and its makers, by structuring that interface, structure how we use the web. In this thesis, I am interested in how various makers of that structure – academic and commercial interests alike – modified it to popularize the web and make it more useful. By grafting the ability to create interactive applications onto the web, browser makers lay a foundation that allowed the web to flourish not just as a universal hypertext but as a new medium for design and a computing platform in its own right.

STANDARDS: SPECIFICATION AND IMPLEMENTATION

While the web became both a design medium and a computing platform, its popularity (and its existence) is in part sustained through the idea that any website should work in any browser. This idea of interoperability is ensured through the development of standards by the Internet Engineering Task Force (IETF) and the World Wide Web Consortium (W3C). "Bit" standards for sending information like HyperText Transfer Protocol (HTTP) and Universal Resource Locators (URLs, which start with "http://") and "syntactic" standards of modeling information like HyperText Markup Language (HTML) are the "building blocks" of the networked world because "[u]sers and platforms have to be able to communicate and interoperate. Otherwise, as far as the network is concerned, they don't exist."[11]

For the web's inventor, Tim Berners-Lee, standards were necessary to enable his "dream for the Web" to be a "much more powerful means for collaboration between people" and further, "a Semantic Web," where "collaborations extend to computers" and "[m]achines become capable of analyzing all the data on the Web – the content, links, and transactions between people and computers."[12] Inspired by Ted Nelson's Xanadu and indebted to ideas such as Vannevar Bush's memex and Doug Englebart's oN-Line System (NLS), in this dream, interoperability was not merely achieved through access but through common meaning, or "syntactic standards."[13]

> A "Semantic Web," which should make this possible, has yet to emerge, but when it does, the day-to-day mechanisms of trade, bureaucracy, and our daily lives will be handled by machines talking to machines, leaving humans to provide the inspiration and intuition. The intelligent "agents" people have touted for ages will finally materialize. This machine-understandable Web will come about through the implementation of a series of technical advancements and social agreements that are now beginning.[14]

This thesis' first case study on the evolution of the web as a design medium will show how this vision was threatened by the methods adopted by browser makers in order to introduce tools for designers to shape the web. By adding support for images and grids in HTML – in spite of the IETF's specifications for the language – programmers of Netscape Navigator (and Mosaic, its predecessor) allowed graphic design practices to exist on the web and accelerated its adoption for commercial uses. However, this advance preceded actions by the IETF and W3C on the development of graphic design for the web, created *de facto* standards for web design, and minimized the authority of the IETF and W3C in shaping the web.

In this case study, I chose to focus on browser makers instead of standards writers because of the gap between specification and implementation. While interactions between members of the IETF and W3C ultimately produce the *specifications* of *de jure* standards, the W3C's "reference edition" – where specifications like HTML+ and HTML 3.2 were implemented in the manner imagined by their editor – was not popular among end users. While Netscape and Microsoft were members of the W3C, they often interpreted these specifications for *implementation* in different ways, addressed gaps in specifications in ways that were visible to users, and occasionally deployed wholly new features not considered when the specification was first written. Limiting the focus of this thesis to the work of standards writers would necessarily overlook the formation of *de facto* standards from users' adoption of particular features and techniques, potentially disregarding those that were not later enfolded into the web's *de jure* standards.

While the JavaScript programming language developed by Netscape is now a standard and an essential part of the web, its formalization as Ecma standard E-262 is a case of a *de facto* standard being submitted for adoption by a standard-setting body in order to create a competitive advantage. To focus my research solely on groups of standards writers would overlook the circumstances of the development and adoption of what is now the world's most popular programming language and the foundation of the idea of the web as a computing platform.

THE OPERATING SYSTEM AND THE COMPUTING PLATFORM

Despite the innumerable mundane uses of the web and its staggering size, contemporary media and technology experts debate whether the web is facing an existential crisis. Smartphone-based platforms and the apps that run on them are displacing the desktop computer and the web browser as the medium by which growing numbers of consumers are using social networking services, mapping, checking the weather, and shopping online. Some say that, through a combination of users' preferences for more streamlined online experiences and consolidation of web traffic on a few sites and platforms created and/or maintained by large companies, the web is dying.[15]

In these debates, the machine-to-machine communication imagined by Berners-Lee and the idea of the web as a computing platform seem natural. Chris Anderson and Michael Wolff, in their *Wired* feature "The Web Is Dead. Long Live the Internet," describe the application programming interfaces (APIs) of sites like Twitter, Amazon, and Google as given and couch smartphone-based apps and web standards as competing implementations. By Anderson's description:

> The rise of machine-to-machine communications — iPhone apps talking to Twitter APIs — is all about control. Every API comes with terms of service, and Twitter, Amazon.com, Google, or any other company can control the use as they will. We are choosing a new form of [quality of service]: custom applications that just work, thanks to cached content and local code. Every time you pick an iPhone app instead of a Web site, you are voting with your finger: A better experience is worth paying for, either in cash or in implicit acceptance of a non-Web standard.[16]

Some frustrated web developers who have grown to prefer making apps for smartphone-based platforms attribute the death of the web to the W3C, accusing them of limiting innovation by forcing browser makers to adhere to a too-limited set of standards. They believe the "web as an application platform stinks" and question the need for browsers to support the same standards because "it limits their innovation, and limits web developers."[17] Others have defended browser makers' process of expanding the capabilities of the web by drawing attention to origins of parts of the newest version of HTML[18] in implementations in new browsers[19], refuting the claim that browser makers are limited in their ability to innovate by the W3C.[20]

This thesis' second case study on the development of the web as a computing platform will address a fundamental misconception at the heart of this debate: the assumption that the web was intended at its invention to be a computing platform. The development of interactivity on the web was executed by Netscape and Sun, motivated not solely by a pure vision of a new platform but by the context of the software industry in the 1990s, a time it seemed that Microsoft Windows would be the world's computing platform in perpetuity.

Personal computing through the 1990s had been dominated by Microsoft, guided by co-founder Bill Gates' vision of "a computer on every desk and in every home."[21] The invention of the web and the founding of Netscape occurred in the 1990s near the peak of Microsoft's reach and Gates' wealth. The alliance of Netscape and Sun Microsystems and Netscape's IPO spurred Gates to direct Microsoft's strategy regarding the internet towards developing its own browser.[22] Until then, Microsoft's monopoly on the operating system and the consumer computing platform was "natural," impenetrable through practices of locking in vendors through Windows' proprietary APIs.[23] By understanding how JavaScript came to be developed, marketed, and ratified as the *de jure* standard Ecma E-262, this case study will show how browser makers' evolving relationship to standards allowed the web to be a computing platform in the first place.

HISTORICAL APPROACH

Past debates about the semantic value of HTML and the role of design were based on faith – not knowledge – that the web would be the primary communication medium of networked society. Contemporary debates have treated the web as a stable technological system, ubiquitous and used naturally. In some ways, the web is a stable technological system – it has been widely adopted and makers of the web create pages and sites that perform adequately on numerous browsers. However, many of the features of the browser that led to the web's adoption are inconsistently implemented, and the fragmentation of the web through different browsers makes work difficult for coders and developers. Recently developed standards like HTML5 and CSS3 and browsers based on WebKit, including Chrome, Safari, and "[a]lmost every smartphone web browser other than Microsoft's," have been challenging the idea that the web is stable.[24]

More saliently, the features of the web that contemporary debates address altered the evolution of the web intended by its inventor, and standards in place to facilitate the web's intended purpose as a global library did not address the potential for the web to become a networked computing platform, made real by Netscape and Sun in 1995. Standards for facilitating graphic design and interactivity on the web were poorly specified if at all, and browser makers' approaches to bending *de jure* standards and creating *de facto* standards in the mid 1990s reflected the same tension between innovation and interoperability addressed today. By taking a historical approach to the process that shaped the modern web, this thesis will situate contemporary debates in a larger story of technological development that involves a variety of social groups.

This thesis will draw from lectures, interviews, email archives, court proceedings, and journalism from the browser wars to describe how uses of the web as a design medium and a computing platform were co-constructed by academic institutions, technology startups, and established firms as well as government officials and graphic designers at the beginning of the modern web. The accelerated software development cycle practiced by browser makers in the 1990s fomented debates among these groups about the role of open standards, dissatisfaction with an existing software development paradigm, and the *raison d'etre* of the World Wide Web. Features of the browser like *view source* and the human-readable designs of HTML and JavaScript gave rise to a new group of expert users whose practices further clouded these debates. Netscape's record-setting IPO was evidence of interest in the web in the marketplace, and the technology startup's naked attempt to erode Microsoft's natural monopoly in the adjacent operating systems market enrolled other established firms like Sun Microsystems to invest in the web – laying the groundwork for the landmark *U.S. v. Microsoft* antitrust case.

Through the SCOT theoretical framework and the browser as the framing technological artifact, this thesis will show how features of the browser – implementations of HTML's *table* element and the JavaScript programming language – moved from interpretative flexibility toward stability. The path of this movement through debates about standards and the software development industry, characteristic of the evolution of the web from its invention as a solution to a knowledge management problem at CERN to its status quo as a "commercial broadcast medium" and part of the infrastructure of today's networked society, is one that echoes today as the internet and people's uses of it evolve.

Chapter 2 Literature Review

This chapter will show how the development of the browser both reflects and diverges from the social construction of technology (SCOT) model proposed by Wiebe Bijker and Trevor Pinch. The SCOT model when applied to technologies like the bicycle, Bakelite, and the Ford Model T describes how their uses are co-constructed by users and designers as an innovation diffuses. This model describes technological development as a "culturally contested zone where users users, patient advocacy groups, consumer organizations, designers, producers, salespeople, policy makers, and intermediary groups create, negotiate, and give differing and sometimes conflicting forms, meanings, and uses to technologies."[25]

Given the history of the browser and the various organizational structures in which it was invented and commercialized, the SCOT model is an appropriate framework through which to understand the roles individuals representing academic institutions, technology startups, established firms, government officials, and graphic designers have played in the development of the browser and subsequent applications of the World Wide Web. Through a list of milestones in the development of the web, this chapter will first define these relevant social groups and describe how they came to be invested in the development of the web. These groups each held unique visions for the development of the web, and this chapter will examine the web's interpretative flexibility as a tension between two broad technological frames: the web as a universal hypertext system and the browser as a ubiquitous interface for a networked society. Finally, this chapter will show how the development de facto and de jure standards for the web's constituent programming languages, primarily HTML and JavaScript, stabilized the web for consumption by lay end users.

THEORETICAL FRAMEWORK: SOCIAL CONSTRUCTION OF TECHNOLOGY (SCOT)

Bijker and Pinch first proposed the model in their 1984 paper "Social Construction of Facts and Artefacts," where they demonstrated it in a case study of the development of the bicycle. In the SCOT model, technology moves from interpretative flexibility to stability and sometimes to closure. Bijker and Pinch defined a technology's interpretative flexibility as the construction of "radically different meanings of a technology" by relevant social groups.[26]

> [T]here is flexibility in how people think of or interpret artifacts … there is flexibility in how artifacts are designed. There is not just one possible way or one best way of designing an artifact.[27]

By identifying "conflicting technical requirements by different social groups," "conflicting solutions to the same problem," and "moral conflicts," the SCOT model allows for the possibility of a range of solutions for these conflicts and problems – "not only technological, but also judicial, or even moral."[28] It also places new "inventions" – and their failures – in a broader context of "stabilizing" existing conflicts. Stabilization, in this model, is a graduated process where a theorist may observe "growing and diminishing degrees of stabilization of the different artefacts" based on a solution's ability to address the range of conflicts.[29] In their case study of the bicycle:

> By using the concept of stabilization, the 'invention' of the Safety Bicycle is seen not as an isolated event (1884), but as a nineteen-year process (1879-98). By the end of the period, the word 'safety bicycle' denoted a low-wheeled bicycle with rear chain drive, diamond frame, and air tyres. As a result of the stabilization of the artefact after 1898, one did not need to specify these details: they were taken for granted as the essential 'ingredients' of the safety bicycle.[30]

Ultimately, stabilization gives way to a state of closure, a point at which relevant social groups shaping a technology determine a problem as solved. The SCOT model allows for either rhetorical closure or closure through the redefinition of the problem. Bijker and Pinch define rhetorical closure in technology as "the stabilization of an artefact" and the perception of the "'disappearance' of problems."

> To close a technological 'controversy' the problems need not be solved in the common sense of that word. The key point is whether the relevant social groups see the problem as being solved.[31]

Stabilization can also give way to closure through redefinition of the problem. In the case study of the bicycle, the air tire solved the problem of vibration for the general public but for sporting cyclists, the air tire was "translated to constitute a solution to quite another problem: the problem of 'how to go as fast as possible.'" Similar events occur in the development of the web as in the case of graphic designers and academic interests through the invention and implementation of Cascading Style Sheets.

The SCOT model was criticized for attributing construction solely to designers and experts, and disregarding the role users played by casting technology as a "black box" given to them. Pinch addressed this criticism in a case study of a relatively stable technology – the Ford Model T – and its appropriation by a group of users – the practice of using its engine as an electrical generator in rural America. He first acknowledged the SCOT model "as originally conceived dealt mainly with the design stage of technologies" and that consequently, "the notion of closure was a little too rigid." "What was missing was a sense of how and in what circumstances the 'black box' of technology could be reopened as it was taken up by different social groups."

Bijker updated the model in his 1995 paper "Towards a Theory of Invention," where he introduces the concept of a "technological frame" and demonstrates it through a case study of the development of Bakelite. He defines a technological frame as "the concepts and techniques employed by a community in its problem solving," which include "current theories, tacit knowledge, engineering practice (such as design methods and criteria), specialized testing procedures, goals, and handling and using practice."[32] The purpose of such a frame is to "structure the interactions among the members of a relevant social group and shape their thinking and acting."

> A technological frame is built up when interaction 'around' an artefact begins. In this way, existing practice does guide future practice though without logical determination. The cyclical movement thus becomes: artefact–technological frame–relevant social group–new artefact–new technological frame–new relevant social group, etc.[33]

This structure, applied to the Bakelite case study, describes why scientists who initially worked with the chemical compounds to formulate plastics had not done so, since they had been working with the compounds under the pretense of producing synthetic dyes. In analyzing the form of the browser, the structure of the technological frame addresses the existence of disciplinary gaps between studies of the browser conducted by members of relevant social groups including computer science historians, business and law scholars, visual culture theorists, and journalists.

Relevant groups and timeline

In the SCOT framework, relevant social groups are institutions, organizations, or individuals who share within themselves "the same set of meanings, attached to a specific artefact."[34] In the development of the web, academic institutions, technology startups, established firms, government officials, graphic designers, and lay end users are broader classes of relevant social groups. As the web grew, lay end users replaced academics as the largest bloc of the browser's users. Browser designers and programmers departed college campuses for technology startups and established firms. Consequently, academics' role in the development of the web changed to one of oversight and management. Government regulators, aside from being (lay) end users of the web, also became enrolled in defining the browser through complaints of monopoly practices by Microsoft in violation of the Sherman Act. Graphic designers who learned to write code in order to practice their craft on the new medium of the web became a class of end user that developed the expertise to make pages and sites.

TIM BERNERS-LEE, CERN, AND ACADEMIC INTERESTS

The web, its browser, and its hypertext markup language (HTML) were invented in 1990 by Tim Berners-Lee as a knowledge management tool at an academic institution: the European Organization for Nuclear Research (CERN).[35] The pre-modern web thrived in academic communities, and two years after Berners-Lee's first demonstration of the World Wide Web at the Hypertext '91 conference, every academic presentation at the 1993 event related to the web.[36] New browsers built on the World Wide Web's protocols included Viola, by Pei-Yuan Wei at UC Berkeley; Lynx, by Lou Montulli, Michael Grobe, and Charles Rezac at the University of Kansas; and Mosaic, by Marc Andreessen and Eric Bina, student employees at the National Center for Supercomputing Applications (NCSA) at the University of Illinois at Urbana-Champaign. Berners-Lee observed that "unlike CERN, NCSA never doubted for a moment that creating commercial products was an appropriate activity," and Mosaic became the flagship product of a technology startup that eventually took the name Netscape Communications Corporation.[37] By 1994, Berners-Lee was struggling to maintain the utility of the web as a universal hypertext in the face of fragmentation brought about by a multiplicity of browsers. With the intention of keeping "a neutral viewpoint" and having "a much clearer picture of the very dramatic, evolving scene than a corporate position would allow," he encouraged MIT and CERN to form the World Wide Web Consortium (W3C) – emulating MIT's X Consortium – to foster collaboration between academics and the burgeoning web software industry.[38]

Netscape as a technology startup

The technology startup whose activities were most salient to this thesis is Netscape, co-founded by Andreessen and Jim Clark, a Silicon Valley entrepreneur who previously founded and ran Silicon Graphics (SGI) which designed and manufactured high-end computer systems. They recruited Bina, Montulli, and other NCSA staff to join Netscape's product development team. This group of young men not far removed from the academic study of computer science was paired with experienced executive personnel, chief among them CEO Jim Barksdale, formerly CEO of McCaw Cellular and COO of FedEx. From its founding, Netscape styled itself a communications company, not a software company. Founded on the belief that "the Web was only the first step toward a networked world. … Netscape's vision did not tie the company to any single product or technology."[39] Their Mountain View, California campus hosted the development of innovations that came to define the modern web including the implementation of the table element in HTML, cookies, and the JavaScript programming language. Clark's decision to take Netscape public before it offered a sustainable business model or evidence of profit proved fruitful. The company's 9 August 1995 initial public offering more than tripled in value – setting a record for an IPO of its size, made Andreessen and some of Netscape's programmers millionaires in a day, and served as an example to other technology startups that they did not need to prove themselves as a business in order to profit on the stock market.[40]

MICROSOFT, SUN, AND ESTABLISHED FIRMS INVEST IN THE WEB

As Netscape's IPO and alliance with Sun Microsystems to support the Java programming language in the browser alerted Microsoft to the growth of the web and the internet as part of consumer computing and their potential to threaten the lucrative monopoly on the operating system Microsoft enjoyed. Microsoft CEO Bill Gates announced on 7 December 1995 – Microsoft's "Pearl Harbor" day that it would abandon its previous internet strategy based on new protocols and televisions and focus on the web. Microsoft and Sun Microsystems represent two established firms whose business models were disrupted by the development of the web, embodied different approaches to the design of the browser.

GOVERNMENT REGULATORS

The U.S. Department of Justice and the European Commission of the European Union in their investigation and finding of Microsoft in violation of antitrust laws saw the browser as a software market that was affected by anti-competitive practices. The National Science Foundation also saw the use of HTML and commissioned the RAND Corporation's 2000 report Scaffolding the New Web, by Martin Libicki et al.

GRAPHIC DESIGNERS

Graphic designers who adapted their skills to the new medium of the web represented another point-of-view. Journal articles by Nalini Kotamraju on the development of the web design profession combine analysis of data including advertisements for job openings and interviews with designers to describe the construction of professional practices. These articles, "Keeping Up: Web design skill and the reinvented worker" and "The birth of web site design skills: making the present history," present the design of the browser and HTML as givens and discounted the effects designers' practices have had on their design.

INTERPRETATIVE FLEXIBILITY: UNIVERSAL HYPERTEXT VS. UBIQUITOUS INTERFACE

As the browser evolved and was adopted, individuals, academic institutions, technology startups, established firms, government officials, graphic designers, and lay end users constructed the web, HTML, and the browser in different ways. This section will look at the design of the web, HTML, and the browser through the composition of two technological frames: the web as a universal hypertext system and the browser as a ubiquitous interface. Berners-Lee's documentation of the web's invention – *Weaving the Web* – and his original proposal for the web, "HyperText and CERN" describe the design of the web to address knowledge management problems at CERN and how that design could scale globally through internet protocols. From NCSA's development of Mosaic, the web began to transition from its academic roots into a medium for commerce, a transition that stabilized with Netscape's IPO and Microsoft's "Pearl Harbor" announcement. As the web's pages and base of end users both grew at an exponential rate, the design of the browser interface became a focus of study for HCI research and popular technology journalism. The concurrent competition between Netscape and Microsoft that redefined the software industry and was the basis for the concept of internet time was an area of focus for scholars of business and law.

UNIVERSAL HYPERTEXT

The web as it was originally proposed was intended to address knowledge management problems at CERN. The laboratory engaged "several thousand people, many of them very creative, all working toward common goals" such as long-term projects like the Large Hadron Collider. While CERN had a "hierarchical management structure, Berners-Lee observed that CERN's "actual … working structure … is a multiply connected 'web' whose interconnections evolve over time."[41]

> In this environment, a new person arriving, or someone taking on a new task, is normally given a few hints as to who would be useful people to talk to. Information about what facilities exist and how to find out about them travels in the corridor gossip and occasional newsletters, and the details about what is required to be done spread in a similar way. All things considered, the result is remarkably successful, despite occasional misunderstandings and duplicated effort.[42]

However, for Berners-Lee, CERN's high rate of turnover led to "information constantly being lost," and "the technical details of past projects are sometimes lost forever, or only recovered after a detective investigation in an emergency. Often, the information has been recorded, it just cannot be found."[43] As a solution, he considered the medium of the book poorly suited to the task due to the size of CERN's organization and projects.

> When a change is necessary, it normally affects only a small part of the organisation. A local reason arises for changing a part of the experiment or detector. At this point, one

> has to dig around to find out what other parts and people will be affected. Keeping a book up to date becomes impractical, and the structure of the book needs to be constantly revised.[44]

Berners-Lee cast as the web's ancestors Vannevar Bush's imaginary memex proposed in the 1945 essay "As We May Think," and Doug Englebart's oN-Line System (NLS), demonstrated in the 1968 "Mother of All Demos." Proposed by Berners-Lee based on Ted Nelson's ideas of hypertext from Nelson's 1965 book Literary Machines and Apple Computer's HyperCard, the World Wide Web was developed with support for multiple protocols in order to allow its users to connect their existing computers to the web and make connections between the various databases of the internet from a single application. By using the internet and its TCP/IP protocols instead of CERN's proprietary networking protocols, researchers at CERN could connect to databases off-site and off-site researchers could share information with each other. He understood the importance of allowing users to continue existing modes of knowledge management – using the workstations they already have – as a key factor for buy-in.[45]

He also saw that CERN was not the only organization afflicted with knowledge management issues and could indeed be "a model in miniature of the rest of world in a few years time," and was unwilling to wait "10 years ... [for] commercial solutions to the problems."[46] Because of CERN's organizational structure, the hypertext system needed to be usable on a wide range of computers including PCs, Macintoshes, Unix, and NeXT workstations. This design made it possible for the system to not just manage information at CERN but at a global scale, and Berners-Lee's aspiration was evident in the name he chose for the system, World Wide Web.

Ubiquitous interface

With the web's protocols established, Berners-Lee presented the World Wide Web to other academics at ACM Hypertext '91 in San Antonio. The presentation captured the imagination of the computer science community, and soon after, new browsers were developed at other academic institutions to provide local access to the emerging global hypertext. Among these browsers was Mosaic at NCSA. Andreessen, a student employee of NCSA, participated vocally in communities formed around message boards and email lists "as if he were attending to 'customer relations."[47]

> This was in total contrast to any of the other student developers. Marc was not so much interested in just making the program work as in having his browser used by as many people as possible. This was, of course, what the Web needed.[48]

Born from this process, Mosaic's software architecture made it the most popular early web browser to an extent that NCSA, in addition to their commercial activities, "was always talking about Mosaic, often with hardly a mention of the World Wide Web."[49] Andreessen, Bina, and other students who built the Mosaic browser left NCSA to start a new company that would eventually become the Netscape Communications Corporation.

Netscape, a technology startup, nakedly recast the browser as a commercial product, and Cusumano and Yoffie's *Competing on Internet Time* studies Netscape as a technology startup in competition with Microsoft. While they analyze the practices browser makers employed while developing products at an accelerated rate, it does not pose those practices against the technological frames of the web's academic forebears and its burgeoning base of users. While Cusumano and Yoffie's interviews reveal motivations behind Netscape's initial public offering and the development of their sales force as a push to make the Navigator browser a ubiquitous interface for networked society, similar analysis of behavior and relationships is not performed on the design of the browser itself.

Samer Faraj, Dowan Kwon, and Stephanie Watts' 2004 paper "Contested artifact: technology sensemaking, actor networks, and the shaping of the Web browser" – a nominal application of actor-network theory to the development of the browser – criticizes the "macro-level perspective that focuses on the industry and the firm as innovation-bearing milieux" and the "micro-level perspective that looks at individual innovation and identifies team-level factors that lead to successful development" for their "limitations when applied to the study of radical innovations such as the Web browser." According to Faraj, Kwon, and Watts, "both streams take a prescriptive and predictive stance, assuming that an optimal innovation process exists, that successful technologies are those that best match the current needs of the marketplace, and that managers can dictate these specifications to their scientists and engineers." While they have framed their study of the browser in actor-network theory, their study suffers from the same affliction as other texts on how browser makers signaled a paradigm shift in the business world. By treating Netscape and Microsoft as actors instead of networks of individual programmers, vendors, and executives who produce software for end users, users and implicated actors (including government officials and established firms like NeXT) play a diminished role in the co-construction of the browser.

Scholars of HCI, through their concern with the user, address in their studies of the form of the browser how well or poorly the interface supports the performance and completion of specific tasks. These tasks, defined by researchers, are determined through statistical analysis of web usage either generated in laboratory settings, diaries of volunteers, or aggregated from other sources. While their studies provide statistics and insight into how browsers are used, by hewing closely to pre-determined tasks, these researchers have only accounted for those tasks for which they have determined meaning.

HCI research on the browser has also glossed over the design of the interface as the product of sociotechnical systems including corporate software firms, open-source software development communities, and standards bodies. Academics in the field including Kaasten, Greenberg, Cockburn, Millett, Friedman, et al have, in theoretical proposals redesigning browser conventions such as history, the back button, and cookies, considered each of these individual conventions of the browser as malleable technologies without strongly considering the technological frames that guided their design. Building on the testing and data collection used as background to their proposals, I will examine how the conventions of designing with tables and JavaScript were designed within the technological frame of the browser as a ubiquitous interface.

Primary data sources will include texts by David Siegel and Brendan Eich. Siegel, whose *Creating Killer Web Sites* was credited with popularizing the technique of designing with tables, wrote in its revised second edition and has spoken at industry events and written in publications with introspection on the effects of the techniques he advocated. Eich, JavaScript's inventor, retains a high profile in the industry as CTO of Mozilla (makers of the Firefox browser) and an active participant in the development of scripting standards on the web.

Stabilization and standards

By the SCOT model's definition of stabilization, in the development of the web, the degree of stabilization was reflected in the process of codifying HTML and JavaScript as de facto and de jure standards. Contention among the IETF community reflected tension between the "dream" of the semantic web, the emergence of the web as a design medium, and consequently "structuralist" and "minimalist" approaches to standards. Libicki et al, under the aegis of the RAND Corporation for the National Science Foundation, reported on the state of web standards in *Scaffolding the New Web*[50]. In their report, they define the minimalists as those who "value simplicity and rapid uptake by the user community. Their standards tend to be expressed as primitives from which subsequent elaboration takes place after acceptance occurs." In contrast, "structuralists value comprehensiveness and precision in the fear that rough-and-ready standards will, at best, grow like weeds, making well-kept ontological gardens that are much harder to maintain. They model the world so comprehensively that no human activity, extant or imagined, would fall outside their construct."[51]

As the "center" of the web moved away from CERN in Geneva and towards Netscape in Silicon Valley, the IETF's participants came to "represent large concerns (some with stratospheric market values)" and Berners-Lee sought to maintain his influence on the web through the establishment of the World Wide Web Consortium (W3C) to "'lead the Web to its full potential,' primarily by developing common protocols to enhance the interoperability and evolution of the Web."

Libicki et al, in observing the role the evolution of the IETF "away from being the progenitor of standards to the body that brings concepts into consensus" and that "HTML and HTTP …

arose from *outside* the IETF," they provide evidence for the diminished role of the IETF and the need for the W3C (outside of changes to the social composition of the IETF consequent to the increasingly commercial nature of the internet spurred by the web). They distinguish the IETF and the W3C as bodies that respectively set the "bit" and "syntactic" standards of the web: while the IETF governs the metaphorical "rails and trains" of the internet, the W3C was founded with the distinct mission of governing the "cargo containers" in which authors and designers placed content in order to produce a semantically rich universal hypertext. They describe five possible ways that syntactic standards could be set without government intervention, each driven by varying degrees of social consensus and software – "letting the market decide," "having different communities each decide," "assuming intelligent software will mediate among various vocabularies," "developing standard ontologies into which standard terms are mapped," and "concentrating on the key words."

The "rough consensus" part of IETF's founding belief of "rough consensus and running code" was generated from mailing lists and discussion boards devoted to various internet standards. Archives of the IETF's *www-talk* and *www-html* mailing lists from 1993-1995 provide rich primary data: Berners-Lee, Andreessen, Hakon Wium Lie (inventor of Cascading Style Sheets (CSS)), Chris Lilley (inventor of the Portable Network Graphics (PNG) file format), Dave Raggett (editor of the HTML 3.2 specification), and other luminaries of the web were active members in these communities. Textual analysis of discussions among these lists' members at key points in the development of the web – the introduction of the img element, the release of Netscape Navigator 0.9 beta – shows a wide range of perspectives over the evolution of HTML and prescience of the web's accelerated growth rate and commercialization.

Divergences

The development of the browser is a unique case to examine in the SCOT model given the relationship between the makers of the technology to the inventors of the technology and the ominous presence of Microsoft, a firm that demonstrated substantial commitments to acquire or destroy whole software markets. On 30 April 1994, CERN announced an open license for the World Wide Web, allowing "anybody to use the Web protocol and code free of charge, to create a server or a browser, to give it away or sell it, without any royalty or other constraint."[52] While in other examples of technology development, an alternative solution to a problem must be substantially different in order to not infringe upon a patent, the intellectual property rights of the web and its protocols gave all browsers a particular baseline of features that they could employ with impunity.

Further, in the case of Mosaic and Netscape, end users played a role early in the development process as Andreessen played an active role in engaging browser "customers." While most companies are reluctant to share information about their products out of fear that other firms will become aware of their plans at an early phase of product development, Netscape's business model involved frequent releases of unfinished products – beta versions – in order to seek end users' feedback and continuously improve their products.

Case studies

Through the SCOT framework, the following case studies will describe the development of HTML and JavaScript through two models of social construction. The first case study on the construction of the page will describe how the table element in HTML was implemented in

Netscape and adopted and appropriated by an emerging group of expert users through features of the browser and their own social networks. Their appropriation of table to introduce graphic design principles in the web played a crucial role in enrolling new lay end users to the web. The second case study will describe how JavaScript, a language designed for a group of users who possessed limited computer programming skills, reflected Netscape's corporate mission. That mission to produce a ubiquitous interface for networked society and its manifestation in the cross-platform browser worked to enroll established firms to the web.

CHAPTER 3 DESIGNING THE PAGE

One of the triggers to the World Wide Web's popular growth in the mid-1990s was Netscape's support for the *table* element in HTML. The *table* element was first supported by Netscape Navigator 1.1, released in March 1995 and the first browser to saturate the market to the degree that the terms "web browser" and "Netscape" were synonymous. Though *table* was part of the HTML+ specification as early as 1993, development of the specification had stalled by mid-1994. By July 1995, months after Netscape implemented *table* and shipped Navigator 1.1, development of a new HTML specification resumed in the form of HTML3. Editor Dave Raggett still considered the *table* element experimental in a "version-zero internet draft" of the new specification. By May 1996, when Raggett's draft took the form of the IETF's *Request for Comments: 1942* (RFC 1942), the web as it was learned by numerous adopters was not merely mediated by a GUI but an aesthetic medium in and of itself. By July 1996, two months after RFC 1942, Tim Berners-Lee declared that "what you actually see when you look at the Web is pretty much a corporate broadcast medium. The largest use of the Web is the Corporation making a broadcast message to the consumer."

How "a few tags for writing physics papers" that Berners-Lee drafted became the *de facto* linguistic infrastructure of a commerce-driven medium is in no small part due to HTML doing things it was never intended to do in the first place. The dichotomy between the IETF's processes of writing standards and soliciting comments and Netscape's accelerated release schedule allowed HTML to evolve in ways that were often driven by corporate interests or personal ingenuity, and key among them was the use of *table* for graphic design. The introduction of graphic design principles into the medium of the web gave the medium commercial utility: web pages could be branded with color, typography, and images composed in two-dimensional space. However, this was accomplished by conflating HTML's structural idiom with its presentational idiom – setting a precedent that favored the development of HTML through proprietary tags by Netscape and Microsoft, which disregarded a "philosophical" foundation of HTML and slowed the development and adoption of web standards like Cascading Style Sheets (CSS) and Extensible Markup Language (XML) in the later years of the decade.

Email transactions between members of IETF's *www-talk* and *www-html* mailing lists – standards writers, academics, commercial representatives, browser programmers, parties generally interested in the novel medium, and designers themselves – in the mid 1990s consisted of debates over the evolution of HTML and the impact presentational markup would have on the web. These debates reveal diverse opinions on the nature of the web and the tension between its academic roots and philosophical foundation and its growing popularity and commercial utility. While agreement that the web was a design medium was emergent, the methods and timetable by which tools in HTML (or other languages) would develop to facilitate modern layout techniques was a point of contention. This point became especially salient as Netscape's browser grew to represent the web's primary interface and the browser's implementation of HTML varied dramatically from the IETF's specification.

While some participants in these debates were remarkably prescient about how designers would misuse HTML, members of the mailing lists did not account for the effects the competitive business context would have on the development of web standards. The competitiveness of the browser market during the early modern web incentivized the development of proprietary HTML, as setting *de facto* standards was a milestone on a path to dominating a market in personal computing – deemed necessary by some executives to ward off elimination by Microsoft. Among technology executives and entrepreneurs, development of the web and the browser presented an opportunity to stanch Microsoft's influence over the computing industry – whether the web remained true to its philosophical roots was secondary to its ability to transform the personal computing industry.

Academic research of this period of time has focused on the formation of the web design profession as a socio-technical system. Nalini Kotamraju's interviews with web designers and studies of how the web design profession was constructed draw attention to the formation of social structures among designers – especially the ways knowledge was disseminated between people through informal knowledge sharing networks. In this chapter, I am interested in the ways in which the data Kotamraju collects reflects patterns of usage by an emergent group of expert users of the web. Further, I will highlight in Kotamraju's compilations of designers' comments how the form of the browser allowed this non-hierarchical method of knowledge sharing to take place, as facilities such as *view source* gave all designers access to the underpinning HTML code of any page on the web.

Key actors including David Siegel, Tim Berners-Lee, and Marc Andreessen spoke and wrote about their contributions to the web and provided commentary and criticism of its evolution. Siegel, a designer whose book *Creating Killer Web Sites* is attributed with popularizing the use of *table* for design, has written about his work in retrospect. In the second edition of his book, he qualified and modified much of the original text to reflect more alignment with the philosophical foundation of HTML. Siegel's criticism of the design techniques he advocated and ownership of the problems they have created reflected an evolving perspective of web design that accounts for ways his work altered the web. Berners-Lee maintained his status as a public figure due to his role as inventor of the World Wide Web and continued to advocate for the principle of a universal hypertext system that he envisioned. Andreessen, as the co-founder of Netscape and a public figure in the birth of the modern web, has in interviews and lectures described the company's relationship to the web, the ways that its initial design is manifest in the company's flagship product, and his early (maker's) perspective of the web's impact on society.

Retrospective texts by makers of the web do reveal some of their intentions and motivations over the design and appropriation of the browser and HTML, but they do little to highlight the particular technological features that connected their designs to the shape of the web. While Siegel, Berners-Lee, and Andreessen each acknowledge the increased role of design and commercial utility of the web, they do not connect this increase to conventions of the browser like *view source*, inventions like the *img* element and other proprietary formatting tags, and ultimately, Netscape's implementation of the *table* that provided designers a set of tools with which to apply graphic design principles on the web.

Through historical analysis of the origins of the table and the effects of its implementation by Netscape, this chapter will highlight the tension between the web's roots in academia and its potential in commerce. Focusing on the rise of Mosaic/Netscape and the development of new HTML, I will show how the people who built a browser that originated at NCSA and eventually became the flagship product of the Netscape Communications Corporation approached the development of HTML as their browser became synonymous with the web. Building on academic research on the construction of the web design profession, I will describe how the budding community of web design came to share information through informal networks and the role the browser's *view source* facility played in the propagation of design techniques on the web. Once I have established the structure of this network, this chapter will describe the ways in which the *table* element came to be appropriated in order to apply graphic design principles to the web. Finally, this chapter will examine the business context of the Microsoft-dominated software industry of the mid 1990s where Netscape operated and that context's role in garnering support among other technological firms for its development of a ubiquitous interface, in spite of the existing standards and processes for developing new standards the IETF had instituted for the universal hypertext of the web.

MOSAIC, NETSCAPE, AND NEW HTML

This section will trace the history of how programmers at NCSA – later Mosaic, then Netscape – added new elements to HTML to allow graphic design on the web through the lens of the IETF's *www-talk* mailing list. On 25 February 1993, Marc Andreessen's proposal of the *img* element to HTML was met with vigorous debate among *www-talk*'s membership, a community of engineers and users of the web.[53] While debate centered on the implementation of such an element – particular syntax of the code and image formats to support – Berners-Lee's response the next day indicated a preference to let the proposal go through IETF's request for comments (RFC) process.[54] Andreessen announced to *www-talk* the implementation of the *img* element in beta version 0.10 of the Mosaic browser two weeks later and the canonical release of NCSA Mosaic 1.0 on 21 April 1993, forgoing IETF's formal process for setting standards.[55] Andreessen and his colleagues at NCSA would later form Netscape and add more design elements and attributes to HTML's vocabulary, including an implementation of the immature *table* element in their Navigator browser that became widely used by graphic designers translating their skills to the new medium of the web.

The growth of HTML's vocabulary and the tools for learning it were defined by Berners-Lee and the IETF in order to establish the World Wide Web as a universal hypertext system and by Netscape in order to establish its browser as the ubiquitous interface of a networked society. For Berners-Lee and the IETF, achieving a universal hypertext system entailed creating HTML from the syntax of the Standard Generalized Markup Language (SGML), separating the content of a page from its design, and setting these standards for browser makers. As the public came to conflate Mosaic with the World Wide Web (and this conflation was exacerbated by NCSA), its programmers' actions jeopardized the universal hypertext system because adding to the HTML specification in only one browser – the most popular one, at that – fragmented the web.[56] For Andreessen and the makers of Mosaic/Netscape, establishing the browser as the ubiquitous interface for a networked society entailed advancing the web in ways that would make it usable to the general public by building support for graphic design techniques into HTML, even if these techniques were not defined as standards for the whole web.

When members of the Mosaic team at NCSA left to form the Mosaic Communications Corporation in 1994, the world's most popular browser became the flagship product of a commercial enterprise headed by seasoned entrepreneurs (Clark and Barksdale). Just as he had as the development of Mosaic advanced, Andreessen announced the release of beta version 0.9 of Netscape's Navigator browser to *www-talk* on 13 October.[57] Like Mosaic, Netscape Navigator included proprietary implementations of new elements that offered coders methods to adjust layout and typography in HTML. Also like Mosaic, Netscape disregarded the IETF's formal process for setting standards. Netscape's additions to HTML in the beta version of Navigator included the *center* and *font* elements which allowed coders to align content to the center of a browser window and choose the typeface and size of a piece of text. Questions about which file formats to support and licensing issues for typefaces and how to render "centered" content in a non-visual browser were either answered without debate through Netscape's implementation or disregarded entirely.

Among members of the *www-html* community, working coders and designers considered these elements welcome additions while other foresighted members worried about how using HTML for designing pages would affect the possibility of creating a semantic web.[58] For Dylan Northrup, a coder and designer at the University of South Florida, the process of making design possible on the web – defining standards for HTML+/HTML3 – moved too slowly. He wrote on 21 October 1994 that the problem was not Netscape's implementation but the IETF's deliberative pace:

> I'm quite sure if the HTML3 group would make some decisions and submit a final draft the guys at netscape would be more than happy to look at it. Perhaps then the HTML3 guys will finally get around to doing some work on HTML3, ne?[59]

Brian Behlendorf, a prominent early web designer who developed the HotWired web portal for *Wired* and is presently a member of the Mozilla Foundation's board of directors, responded a few hours later that the HTML+ standard, had been developed and ready for at least a year, and that the greater concern was not that Netscape "chose not to implement any of it as an experiment" but that "they didn't even express any opinion about it at all."[60]

Part of Netscape's impatience was due to the competitiveness of the browser market, but their continued development of formatting elements and attributes was at least in part due to their customers' demands for control over layout and design. Unfortunately, the way Netscape satisfied customers' demands threatened to undermine the semantic power originally designed into HTML. Behlendorf argued that "dangling the carrot of page-layout to document authors" was a slippery slope, and that "adding tags every once in awhile to approach a page layout language isn't going to work" – effectively espousing a structuralist perspective towards the development of web standards for design.[61] Northrup's response argued that as long as additions to HTML were "needed/useful," "what's the problem?"[62]

This argument about Netscape's redefinition of HTML reflected what Behlendorf called the "social problem" of giving users of the semantic markup language the ability to use it also as a page layout language.[63] Coders given some control over design would not be satisfied, and Behlendorf accurately foretold of other non-semantic extensions of HTML. For Berners-Lee, the development of proprietary HTML was a byproduct of the existence of more than one browser, especially one created with commercial motives.

> Whenever one browser had a feature, an adaptation of the protocol, and the other one didn't, there was a possibility that the other one would adapt, would create that feature but use a slightly different syntax, or a very different syntax, or a deliberately different syntax. You get places where you find a little message which says, 'this page has been written to work with Mosaic 5.6 or NetScape 3.0 or Internet Explorer 2.8 or whatever it is, and it's best for you to use that browser.'[64]

Berners-Lee attributed his decision to start the W3C to the fragmentation of the web brought about by Mosaic and perpetuated by Netscape (and later, Microsoft). The W3C, established to preserve the possibility of machine-to-machine communication over a semantic web, oversaw the web's *syntactic* standards as opposed to the *bit* standards – the protocols for exchanging information – under the purview of the IETF.[65] In 1994, overseeing the web's syntactic standards meant stabilizing and controlling the growth of HTML's vocabulary.

The W3C's mandate to give "a comprehensive structure to the universe of documents" and structuralist approach to writing standards was at odds with the initial design of the web, which according to Libicki et al "rose to prominence as a set of well-chosen primitives [(the first HTML elements and design based on SGML)] "rather than the expression of any such structure."[66] For browser makers, support for a given type of HTML rendering – fragmenting the web – became a way to create a competitive advantage. Through 1998, Netscape continued extending the language in ways that tacitly encouraged designers to continue using structural markup for visual design, ensuring that their designs relied on Netscape's proprietary vocabulary – and Navigator web browser – to be properly experienced, including *frameset* and its constituent elements, the *center* element, the lamentable *blink* element, and multimedia support through the *applet* and *embed* elements. Netscape for its part continued to develop HTML tags for document presentation in the paradigm that followed the misuse of *table*. In addition to the now-deprecated *multicol* and *layer*, Netscape's introduction of the *spacer* in Navigator 4.0 (addressing the method of using a "blank" image in order to create white space in table-based layouts) implicitly validated the method of designing web pages with structural markup – *spacer* only had presentational value and no structural value whatsoever. The web-as-aesthetic-medium was the new paradigm, the web-as-marketplace was taking shape, and the web was largely optimized for Netscape. These elements were adopted by coders in their designs in part due to the *view source* facility provided by browsers and the makeup of the new field of web design.

Better learning through *view source*

This section will describe how coders and designers formed the web design profession and acquired skills through features of the browser like *view source*. By revealing the literal source code of any page on the web, *view source* is a facility that effectively allows the browser to reveal to its users the tools by which they can become designers. The readability of HTML – based on SGML – made it more accessible than other programming languages, the presence of tools to write and test HTML were applications included in most operating systems, and *view source* gave users of the web innumerable examples of markup in use that they could copy and tweak. While it made "the entire web into a giant learning lab,"[67] it was a lab where there were no best practices against which a user could compare the code they learned because the budding field of web design had few institutions. Consequentially, non-standard HTML code can perpetuate if its effects are seen as desirable. With this practice of using *view source* to learn to create pages, the additions Mosaic and Netscape made to HTML became as widespread as their respective browsers' base of users.

Because new coders increasingly came from professional communities outside of computer science – in particular, graphic design – they demonstrated greater concern with HTML's properties as a layout language than its properties as a semantic markup language. From Kotamraju's fieldwork on the formation of the web design profession by employers and practitioners, at least one designer she interviewed explicitly identified *view source* when "emphasiz[ing] the self-taught nature"[68] of the skill of coding HTML:

> There's a lot of resources online. And there's a lot of books. And I think probably the most common way of learning, maybe not learning the basics, but learning new things is probably by looking at the sources of pages . . . They see something that is cool and they

> want to know how to do it. It's right there, you just have to *view source* and then you can see exactly how it was done.[69]

Among web designers, using *view source* to see a page's underlying source code and learning techniques by copying, pasting, and tweaking source code is an acceptable and often recommended practice. As web designers tended to advance their skills through what sometimes amounted to trial-and-error, knowledge sharing networks among web designers came to take to form of "colleagues, instructional books, or professional groups online … more so than formalized education … and traditional professional institutions did not play a large role in shaping the web design skill-set." Professional institutions that eventually emerged proved "impotent," and they "did not seem able to, or even interested in, regulating and freezing the web design skill-set."[70]

The emerging web design profession was also marked by a division between designers and non-designers that made the ability to code paramount. Web designers were not uniformly rooted in graphic design backgrounds; competent coders who could execute in HTML competed for the same work. This not only had ramifications for design on the web but in the design industry in general.

> Embracing the necessity to code would allow non-graphic designers to exercise their claim over web design and possibly weaken the graphic design community's claim to web design, but also perhaps to the expertise of design itself. The tension of web design's existence between art and code opened the door to competing claims of what knowledge should be part of the skill-set and who holds this knowledge.[71]

However, the graphic design industry "denigrated web design as inferior to genuine visual design." "Trade publications … hint[ed] at contentious debates" over whether web design was "art or code," and "often characterized web design as a form of design limited intrinsically by technology, an instance of code dominating art." Designers who held this view called the web "a clumsy environment to work in, and severely limited in graphic control," and Kotamraju observed:

> Even the computer magazine, *MacWeek,* points out the limitations of programming, "[HTML] is a limited language that doesn't give designers free reign in their attempts at creative Web construction."[72]

Whether or not writers at *MacWeek* and others in web designers' networks had confused HTML for a layout language, one of the web's selling points was its nature as a GUI for the internet. Graphic design would necessarily have to exist on the web, and graphic designers who learned HTML comprised the majority of the new profession of web designers. Among the new professional community (which included user interface designers from computer science backgrounds) coding expertise was a requirement, in addition to knowledge of other limitations like the bandwidth of dial-up internet connections and the size and resolution of computer monitors. At the Second International World Wide Web Conference (*Mosaic and the Web*) in October 1994, Christine A. Quinn presented the first recorded instance of employing "both a graphic designer and an art director" to implement "a concept that brought architecture and color to [the Stanford Center for Professional Development]'s World-Wide Web catalog."[73] In her presentation, Quinn sidestepped debates over art and code and instead described the process employed by the designer and art director as they were introduced to the web. Given the addition of the computer monitor as another layer of the medium with its own characteristics, "[i]t wasn't simply a matter of creating a piece of art and turning it into a .gif file." She described "the artist, unfamiliar with the new requirements of the medium" as having "no idea how to proceed" and "the art director" as "able to instruct her on the need to keep the screen in mind and to apply such techniques as anti-aliasing to the text."[74] Although the demand for web design grew, the novelty of the web meant that the supply of designers native to the medium was scarce, and the standards for designing on the web were being set without strong governance. Graphic designers whose skills had been honed for print were learning to code HTML before the CSS technology was even drafted let alone successfully implemented in a known web browser, and designers were keen to use any code, however standard or proprietary, as long as it was effective.

HTML, due to its origins as a semantic markup language, was not easily adapted to a page layout language that offered the degree of control to which graphic designers were accustomed. Graphic designers' apparent disregard for the web as a visual medium had created a rift within the industry, and the spectacular growth of the computing sector made web design an attractive career choice despite this. Finally, the use of *view source* for ad-hoc learning and informal knowledge-sharing networks upon which web designers relied meant that coding practices, however standard or proprietary, would be perpetuated if they achieved a desired result. The browser's accelerated software development life cycle made it difficult for designers to maintain a current awareness of HTML, making later advances towards interoperable standards difficult to synchronize. These factors that defined the web design industry during the early modern web provided a dry field for Netscape's new proprietary formatting elements to scorch. The element that would be the spark and give the budding web design community the opportunity to apply formal practices of the graphic design to the medium of the web would be the *table* element.

Reinventing the table

Building on Mosaic/Netscape's activities as the makers of the world's most popular browser and the knowledge-sharing practices of the growing community of web designers, this section will describe how Netscape's implementation of HTML's *table* element became a foundation of the early-modern era of the the web (defined by the influx of commercial interests). While in 1994, Netscape's introduction of new HTML elements for design had drawn the ire of the IETF *www-html* and *www-talk* communities, their implementation of *table* and its constituent elements made it possible for coders and designers to directly translate the principle of the grid from modern graphic design to the medium of the web. Advances in digital desktop publishing manifest in PageMaker and QuarkXPress meant that control over color and typography afforded by recent proprietary HTML elements like *font* would no longer be sufficient: designers needed a way to control the two-dimensional space inside the browser window. Designers working in the nascent medium learned through informal knowledge-sharing networks and *view source*, and several credited David Siegel's *Creating Killer Web Sites* and his work with Studio Verso with popularizing techniques for applying graphic design principles on the web through HTML tables. However, Siegel's book propagated the use of hacks to affix layouts regardless of the user agent, defined principles that software developers used to create WYSIWYG web development programs, and the technique, in Siegel's own words, "ruined the Web."[75] Through Netscape's interpretation and defiance of *de jure* standards, pages could be branded with color, typography, and images composed in two-dimensional space, giving rise to commercial uses of the web.

Among graphic designers working on the web, the *table* element made work guided by grids possible. Modern design has been characterized by relationships to grids – systems of laying out two-dimensional space in equally sized and spaced blocs – and the use of grids was meant to create a sense of clarity and credibility. In Mosaic and Netscape, designers could place images alongside text and were able to make slight modifications to typography. Before the *table*, design on the web was a strictly linear endeavor, and for that, arguably not design at all. The grid system, "by creating a sense of compact planning, intelligibility, and clarity," "suggests orderliness of design," "lends added credibility to the information and induces confidence."[76] In the context of the web, that websites were cheaper to produce than printed collateral made it possible for even smaller companies "to look as large as a big company and be as accessible as a big company on the Web."[77]

> Big companies spend hundreds of millions of dollars building their distribution channels. And the Web is going to completely neutralize that advantage.[78]

A small company's ability to use a website as part of a last-mile solution in their distribution channel – replacing expensive brick-and-mortar locations with a cheaper modern, credible e-commerce website – made the web "an incredible democratizer."[79] In 1996, HTML's *table* element and its children were the only means by which designers could compose two-dimensional pages and as such were used to that end, in spite of HTML's intended design.

HTML was not intended to be a page layout language – as a spinoff of SGML, HTML was intended to give structure to documents, and it remained the task of the web browser to format this structure as it saw fit. In RFC 1942, Raggett wrote "[a] major consideration for the HTML table model is that the fonts and window sizes etc. in use with browsers are not under the author's control" – explicitly reinforcing the structural nature of HTML as other elements had before it.[80] Cascading Style Sheets, proposed by Hakon Wium Lie, were being adopted by the

IETF and World Wide Web Consortium (W3C) to address the issue of author-controlled formatting; in the meantime, web browsers were supposed to format documents adequately as the markup demanded.

However, RFC 1942 was drafted more than a year after Netscape had implemented the then-experimental *table* element, and by then, Siegel and other designers learned how to use it for graphic design. Netscape's programmers and later Raggett modeled HTML tables "to allow simple tables to be expressed simply with extra complexity only when needed. This makes it practical to create the markup for HTML tables with everyday text editors and reduces the learning curve for getting started."[81] By using the HTML table as a page grid, graphic designers from print backgrounds could work in the medium of the web without altering their skillset to incorporate the paradigm of semantic web design intended by HTML specifications. A primer on web design published in 1998 described their use:

> HTML's table layout tools were designed with traditional tables in mind – the kind that hold statistics within a page – but designers quickly adopted them for structuring whole pages. With tables, designers can specify different-width columns to break up a page horizontally and discrete rows to control vertical space. [...] [T]ables do allow designers to specify, to the pixel, exactly where text or images will be placed on a page – a control they never had before.[82]

Once coders had the ability to "specify, to the pixel, exactly where text or images will be placed on a page," designers could operate with the same precision to which they had been accustomed in desktop publishing software.

When Netscape practically held a monopoly on the browser market, this worked – designers created visually compelling work using HTML tables and "spacer" images – and *Creating Killer Web Sites* became a best-selling book translated into ten languages. The program NetObjects

Fusion, based on Siegel's principles – and often his exact HTML code – took "a baby step toward becoming the first PageMaker of the Web."[83] In doing so, Siegel's work furthered Netscape's proprietary implementation of HTML and reinforced the fragmentation of the web. While in Siegel's estimation, he "ruined the web by mixing chocolate and peanut butter so they could never become unmixed," referring to "the hangable offense of mixing structure with presentation,"[84] the separation of structure and presentation was a philosophical foundation of the web that would allow for the possibility of machine-to-machine communication.

While Siegel defended the first edition of his book and his early work with Studio Verso with the declaration that "the Web is a visual medium – not to design is to design," he failed to account for the non-visual uses of the web.[85] Raggett's introduction to RFC 1942 highlighted that "[table data] cells can [...] include attributes assisting rendering to speech and Braille, or for exporting table data into databases": applications of HTML tables that are not visual and rely on the separation of structure and presentation in order to be read.[86] Although Siegel joked that because of his methods "search engines can't tell a picture of Dolly Parton from a picture of Dolly the sheep," this scenario would not have been an unusual consequence given that search engine indices were formed by non-visual browsers that "crawled" the web. Siegel conflated his branding as an "HTML terrorist" with his desire to advance the visual experience of the web. However, his act of "terrorism" was not his advancement of the visual experience of the web but his advancement of the idea that the web is solely a visual medium – an idea that allowed him and other designers to mix structure and presentation with impunity from all but web standards evangelists. Siegel's myopia – evidenced by his failure to include the adverb "usually" in his assertion – coupled with just enough coding expertise to be dangerous enabled other graphic designers to engage with the web through this paradigm before the tools in the language were created.

Siegel and other designers who addressed gap between the web's potential as a graphical design medium and the lack of two-dimensional formatting control by misusing HTML's *table* element as a design grid evoked Chris Lilley's prescient caution that "if style sheets or similar information are not added to html, the inevitable price will be documents that only look good on a particular browser, at a particular window size, with the default fonts." As early as May 1994, Lilley, creator of the Portable Network Graphics (PNG) file format and one of the first to academically explore the graphic qualities of the web, recognized that support for inline images on the web was a step towards making it a design medium for commercial interests. Siegel, in an attempt to distribute blame assigned to him by "HTML and XML purists," branded Andreessen the "founding father of the HTML Terrorist Guild, which now numbers in the thousands" for his development of HTML's *img* element.[87]

> Andreessen came up with the idea of the <IMG> tag [...] the Web was both born and destroyed at that moment.[88]

Like Siegel, Lilley supported advancing the visual experience of the web and thought that when "images were allowed inline in HTML documents, the web became a new graphical design medium."[89] Taking into account that people who want "commercial sites want a corporate look, an image, and are prepared to pay real money to get it"[90] and that "these people are, at least, a sizeable miniority (sic)," Lilley declared that "there should be a means for them to achieve their ends" and warned of the same "inevitable" fragmentation as Berners-Lee. Siegel defended his approach:

> Here in the spring of 1997, I still have to use what works. My clients want to win on the Web, so we employ the method used by more political strategists: Image. We use great-looking sites and compelling experiences to create equity on the Web for our clients. … Amazon.com will have a lot of competition. All of them will have great selection, service, and advice on what to buy. The battle will be fought on design and editorial content, plus extra services that make people feel special. This has nothing to do with "information," but everything to do with attracting and keeping customers. How much information does Nike give out about its products? Not a lot. On the commercial side of the Web, design can make millions of dollars of difference.[91]

The connection of design on the web to the creation of "equity" made the web a compelling medium for graphic designers, and by 1996 the "Web [was] pretty much a corporate broadcast medium. The largest use of the Web is the Corporation making a broadcast message to the consumer."[92]

The fact that the web's exponential growth rate was mirrored in the user base of Netscape Navigator further complicated the unmixing of structure and presentation. New web users who adopted the point-and-click GUI-driven web required websites to be GUIs too, and in Netscape's singular implementation of the *table* element, designers found a way to deliver GUIs to users, often on behalf of commercial interests. The *table* element and its consequent (mis)use prescribed by Siegel and perpetuated by other designers through informal networks and *view source* became Netscape's killer app: facilitating the graphic design of web pages. Perpetuating the cycle, it became necessary for other browser makers to adopt in order to maintain interoperability with sites created for Netscape, a number that grew exponentially.

The continued use of HTML tables for page layout by new web designers not only made a return to the ideal of semantic and accessible markup difficult but emboldened Netscape to advance its flagship program in ways that disregarded the emerging standards of HTML 3 and CSS. While disregard for open standards was sometimes poorly perceived among programmers, other Silicon Valley executives expressed their support Netscape's practices for the more salient purpose of outflanking Microsoft's monopoly on the personal computer platform.

DEFENDING AGAINST MICROSOFT

This section will describe how Netscape's proprietary additions to HTML, however harmful they were to the ideal of the World Wide Web as a universal hypertext, were rationalized as part of a larger strategy to contain Microsoft's grasp of the internet. By the time Mosaic became Netscape Communications Corporation and Andreessen became the company's public face on the covers of *Time* and *Fortune*, Microsoft's Windows and DOS operating systems were installed on 80% of computers worldwide and Microsoft was among the world's dominant technology firms and one of the five largest companies in the world.[93] Chairman and co-founder Bill Gates was the world's richest man and he had in the process of growing Microsoft made enemies among his competitors and cultivated a negative public image. Microsoft had not yet invested in the web and as the web grew, some computing companies saw in it the potential to loosen Microsoft's monopoly on the operating system layer of the personal computing platform.

Heads of some of Silicon Valley's largest companies rallied around Netscape both implicitly and explicitly. For some, loosening Microsoft's monopoly through the web was more important than getting the web 'right.' While Berners-Lee envisioned a universal hypertext system, CERN's open license to develop web servers and browsers meant that Microsoft could easily enter these markets. In order to prevent Microsoft from dominating another facet of personal computing, executives of other companies saw in Netscape's domination of the browser market an opportunity to restore competition and spur innovation in consumer computing markets.

In a 1996 interview, NeXT CEO and Apple co-founder Steve Jobs observed that "[t]he most important thing for the Web is to stay ahead of Microsoft … anything that slows down the Web reaching ubiquity allows Microsoft to catch up. If Microsoft catches up, it's far worse than the fact the Web can't do word processing." Adding "those things can be fixed later," Jobs' desire to keep Microsoft from gaining a foothold on the web functioned as an implicit advocation of Netscape's work to popularize the web, even if the company's browser did not adhere to the web's open standards.

> Microsoft's proprietary history offered Netscape a powerful lever. Microsoft could never afford to put its existing technology into the public domain. Opening Microsoft's APIs and device drivers would destroy Microsoft's business model. As Lou Montulli told us, "Microsoft is the antithesis of the Internet, or at least the way the Internet was. Microsoft has changed a great deal, but there is, and probably will always be, this suspicion in people's minds. There still is a large anti-Microsoft contingency, [and] we benefit from it to some degree." In fact there are thousands of disaffected software developers and Microsoft competitors looking for alternatives to Microsoft. These companies, often called the ABM crowd, "Anybody But Microsoft," were a natural constituency to align with Netscape.[94]

Sun Microsystems was led by Scott McNealy, who envisioned "the day when disposable word processors and spreadsheets will be delivered over the Web via Java, priced per use."[95] More pointedly, McNealy saw in his company's Java programming language and Netscape's browser a way to outflank Microsoft's monopoly over the operating system, to "blow up Gates's lock and destroy his model of a shrink-wrapped software that runs only on his platform."[96] This vision was formalized in Sun's partnership with Netscape in 1995 to leverage the browser to distribute Java applets.

Within Netscape, co-founder Jim Clark had previously founded high-end computing firm Silicon Graphics and ceded that market to Microsoft and Intel. From the experience, he learned that the best way to defend against acquisition or attack by Microsoft – "to preserve [one's] station in [Silicon] Valley life" – "was to create a monopoly."[97]

> In computing, the monopoly took the form of a toll booth. Bill Gates had his own toll booth, the PC operating system. Jim Clark wanted his own toll booth.[98]

Although Clark and Andreessen observed that open standards "are not merely desirable" but "essential" in a networked world, Netscape ultimately shared with Microsoft a business model founded on ubiquity, and ubiquity based on a *de facto* proprietary standard was better business than being absent from the market.

> The key to success for the whole thing was getting ubiquity on the [browser] side …. It's basically a Microsoft lesson, right? If you get ubiquity, you have a lot of options. … You can get paid by the product that you are ubiquitous on, but you can also get paid on products that benefit as a result. One of the fundamental lessons is that market share now equals revenue later, and if you don't have market share now, you are not going to have revenue later. Another fundamental lesson is that whoever gets the volume does win in the end. Just plain wins.[99]

While Netscape and the web reached public consciousness in 1995, Microsoft led the market for all computer software and had not yet developed a web browser of its own.[100] It had originally assigned five programmers to develop Internet Explorer, but on "December 7, 1995, … Gates gave a speech to his employees outlining Microsoft's aggressive new approach to the Internet. He named Netscape as a target and rallied a team of top-notch programmers to build Internet Explorer."[101] The team that developed version 3.0 of Internet Explorer grew to 300 programmers and, as Microsoft sought to grow its share of the web browser market at the expense of Netscape, each company extended its browser's HTML support with its own proprietary tags.

New browser versions that broadened the practice of web design were released at a pace that the IETF and W3C could not match. In the IETF's process, a proposal's evolution into a draft and then a standard ideally took an "expeditious" ten months: "six plus months for the Internet community to comment on proposed standards … and four plus months more until actual promotion to a standard."[102] That timespan grew as participation in the IETF grew – attendance at the IETF's semiannual meetings grew from 15 in 1986 to 100 in 1987 to 2,000 in the 1990s.

> The growing crew of network designers, operators, vendors, and researchers [interested in the web] collectively created a bottleneck. … Between 1993 and 1999, it took roughly 3 years for a proposed standard to become a draft standard, and 5 years for a proposed standard to become a standard.[103]

In the context of "Internet time," three years was too long. The growth of the web doubled the size of the internet every 53 days.[104] Netscape's ethos of shipping products quickly and competitive pressure from Microsoft brought Navigator 2.0 to launch in March 1996; both companies shipped version 3.0 of their respective web browsers in August of the same year. By then, HTML 3.2 was still more than a year away from recommendation by the W3C.

CONCLUSION

If sheer growth is the measure of success, the web benefitted from Netscape's outsized programming acumen. Today's web is usually a visual medium and corporate websites are branded equally to other forms of visual collateral. The group of people comfortable using the web today is far broader than the CERN physicists who Berners-Lee considered colleagues. The rapid diffusion of the web in the mid-1990s – doubling the size of the internet every 53 days –

may have been delayed or stunted had the web's linguistic infrastructure not been so misused by both software developers and designers.[105]

As Berners-Lee was credited with inventing the World Wide Web and its "well-chosen primitives,"[106] his vision of the semantic web where machines could "become capable of analyzing all the data on the Web"[107] was lost in Netscape's rush to enroll users and designers into the web. Mosaic's makers at NCSA, particularly Andreessen and Bina, had a reputation for adding elements to HTML that were either experimental in nature or wholly new. These elements, like *img*, *center*, and *font*, addressed designers' demands to add basic formatting to pages on the web.[108] However, they also changed HTML from "an abstract, structured, markup language where authors marked the different logical roles of the text (paragraphs, headlines, lists and so forth) towards a concrete presentation language where emphasis is on the final form presentation of documents (fonts, colors and layout)."[109]

Web designers, who aside from constructing pages on the web were constructing their own profession, learned design techniques through informal networks and the browser's *view source* feature. This method allowed techniques like using HTML tables as design grids to proliferate rapidly, and Netscape's popularity in the browser market meant that other browser makers would have to follow suit in order to render designed pages. In Netscape's method of introducing the *table* tag and Siegel's leverage of a half-cooked technology to introduce compelling graphic design to the web, the web's structural idiom became mixed with its presentational idiom, inadvertently sabotaging the web's original linguistic intentions. However, as the web grew increasingly popular, commercial, and associated with Netscape, the Navigator browser and its proprietary extensions of HTML took on another use to technology executives: more than a bulwark against Microsoft's grasp of personal computing, it could free the computing platform from Microsoft entirely.

CHAPTER 4 CONSTRUCTING THE PLATFORM

In April 1995, Netscape Communications Corporation hired Brendan Eich, and in ten days in the month that followed, Eich invented what is now the most popular programming language in the world. Named JavaScript, it was designed to "address the lack of programmability" of web pages, which were largely academic documents comprised of static text and occasional graphics and in any form a novelty to the general public.[110] Eich and Andreessen "envisioned a much livelier Web consisting of pages that acted more like applications," and Eich's goal in the design of JavaScript was not only to produce "a scripting language … that could touch elements of the page, change their properties, and respond to events" but one that was easily learned and modularly implemented.[111] What Eich presented to Netscape's management as a demo became a "fait accompli," and the language was first supported by Netscape Navigator 2.0, released as a beta on 10 October 1995 and shipped to the public on 5 February 1996.

Unlike the *table* element, there was no standard drafted for interactive websites when Netscape implemented JavaScript in their browser. Instead, Netscape through a combination of internal motivation and outside partners invented and implemented JavaScript and other features in the browser that threatened the position of the operating system in the consumer computing platform. JavaScript's design, inspired by obscure programming languages – Self and Scheme – and the modular model of Apple's HyperCard, made it incredibly robust and usable by amateurs. Despite this, computer science literature on JavaScript has tended to be either instructional or derisive – treating the language as fully formed or inferior rather than analyzing the process that led to its popular and robust design. Netscape advertised that JavaScript would enable the web to be more than pages of text and a platform for fully interactive applications. Netscape's submission of JavaScript to Ecma for standardization and Microsoft's decision to reverse-engineer the language to create JScript in Internet Explorer made the language both a *de jure* and *de facto* standard for scripting on the web. Business and law texts on the browser wars and U.S. v. Microsoft place the invention and design of JavaScript in the background of their analysis rather than tracing the narrative between them and executive decisions broadly related to computing as a platform.

Through analysis of the development of JavaScript by Netscape, its distribution as a *de facto* standard, and adoption as a *de jure* standard, this chapter will show how defining features of the modern web browser were designed to position the browser as the ubiquitous interface for a networked society. I will place accounts of the invention of JavaScript by Eich and Crockford in the context of Netscape's business strategy to show how the language's flawed design, while criticized by programmers, also accomplished certain executive goals. By juxtaposing business literature on Netscape's cross-platform design strategy, media coverage of the then-novel web, and the company's proposal of JavaScript for standardization by Ecma, I will show how Netscape's attitude towards open standards evolved to reflect the heightened level of competition in the browser market consequent to Microsoft's entry. Finally, this chapter will situate JavaScript in the broader development of the web as a computing platform, a development that thrust a young company strained by rapid growth into a "format war" against a formidable competitor.

Computer programming in the public sphere

While the development of the web as a computing platform has been discussed in the context of economic events (the browser wars) and a monumental legal case (*U.S. v. Microsoft*), literature from business and law have placed the invention and design of JavaScript in the background of their analysis, instead focusing on executive decisions broadly related to computing as a platform. Given the role programmers played at the executive level of software companies, economic analysis and mainstream media coverage of the birth of the web have included interviews of Andreessen, Eich, and others. By placing engineers in the foreground of public discourse about the web, the process of technological development is spotlighted and the relationship between the executive mission and computer programming is laid bare. Netscape's rise to prominence – their association with the growing web and record-setting IPO – was frequently a subject of journalism by mainstream publications including the New York Times, Wired, Vanity Fair, and Fortune. While upper management largely represented the company in these publications, key technical personnel including Eich were also interviewed and profiled in the narrative of the evolving web, though their work as programmers and designers caters to broader (less tech-savvy) audiences. On the other side, computer science literature on JavaScript has tended to be either instructional or derisive – treating the language as fully formed or inferior rather than analyzing the process that led to its popular and robust design.

Eich, through his continuing role in the browser programming as Chief Technology Officer of Mozilla (makers of Firefox) and ongoing involvement as an editor of the EcmaScript specification, is a public figure in the global JavaScript development community. He and Douglas Crockford, Yahoo! JavaScript architect and the inventor of JavaScript Object Notation

(JSON), both often speak on the history and design of Eich's most widely used invention. Interviews with both Eich and Crockford each comprise a chapter of the 2009 book *Coders at Work: Reflections on the Craft of Programming* by programmer Peter Seibel. These interviews, as conversations between programmers, include critical reflections on their respective contributions to the birth and development of JavaScript and EcmaScript and the language's ancestry and legacy in computer science.

This relationship between Clark's business experience, Netscape corporate mission, and design strategy forms the foundation for this chapter's analysis of how the browser – designed to run on multiple existing computing platforms as a manifestation of Netscape's mission – itself came to be a computing platform. Besides the unorthodox timing and outsized success of their IPO, other Netscape corporate practices – and Microsoft's reactions to them – were analyzed in popular business literature and academic texts by business and law scholars. Michael Lewis's *The New New Thing*, a biography of Clark, described how Clark's prior experience battling Microsoft and his own board of directors at Silicon Graphics affected how he structured Netscape as a business (giving ownership shares of the company to Andreessen, Eric Bina, and other NCSA alumni) and came to emphasize ubiquity as a defining feature of Netscape's business plan. However, Lewis is less concerned with the relationships between Clark's business experience, Netscape corporate mission, and design strategy and the development of the web as a computing platform. That Netscape Navigator was designed to run on multiple existing computing platforms is a manifestation of Netscape's mission, broadly described in Michael Cusumano and David Yoffie's *Competing On Internet Time*. Their book probes the subject of Netscape's business practices and how they evolved as the market evolved as Microsoft's internet strategy shifted to the web and release of the browser. Their interviews with key Netscape personnel tied the company's broad mission to their strategy to develop Navigator to run on Windows, Macintosh, and Unix platforms and their decision to align with Sun to support Java in the browser.

In his seminal talk "What Is Web 2.0?" Tim O'Reilly compared Netscape's and Google's definitions of the web computing platform. While he placed Netscape in the company of "Lotus, Microsoft, Oracle, SAP, and other companies that got their start in the 1980s software revolution," he contextualized Google's definition of the platform from its beginnings as a "native web application." As a native web application:

> None of the trappings of the old software industry are present. No scheduled software releases, just continuous improvement. No licensing or sale, just usage. No porting to different platforms so that customers can run the software on their own equipment, just a massively scalable collection of commodity PCs running open source operating systems plus homegrown applications and utilities that no one outside the company ever gets to see.[112]

In between O'Reilly's characterizations of the two companies is browser makers' relationship to the concept of a *native web application*, and Netscape's role in particular as a part of the "old software industry." Once-proprietary technologies like JavaScript, cookies, and XmlHttpRequest form the computing platform upon which native web applications that characterize Web 2.0 are built. It is ironic that these proprietary technologies rendered the "software licensing and control over APIs" that characterized their invention as irrelevant: they made it possible for "software never need[ing] to be distributed but only performed."[113] While the browser itself represented "an acceptable lowest-common denominator interface across different platforms"[114] and offered limited options to programmers who built applications for the web instead of desktop computers, the web was appealing as a distribution channel and its constituent technologies granted coders adequate control while being easily learned.

GLUE LANGUAGE

This section will describe the background and context at Netscape that informed the design of JavaScript. Andreessen's vision for Netscape included developing a method for amateurs to program for the web, and Eich's computer science background informed the robust design of JavaScript. Esteem of Andreessen as a visionary and Eich as a computer programmer earned Andreessen the support of investors and Eich the trust of Netscape's management. With this latitude, Netscape hired Eich to develop a wholly new programming language for the browser rather than attempt to support an existing programming language natively on the web. The new language – originally named LiveScript, then Mocha – was designed to be learned "by the yard," meaning an amateur programmer only needed to know a few pieces of the language in order to add interactivity to pages and sites.

However, JavaScript was too hastily released, and bugs in the language were exposed and exploited by unscrupulous programmers. This earned JavaScript a negative reputation among programmers, who have also considered it a "little brother" to the "real" Java programming language. The browser's *view source* facility and lack of a compilation step[115] and JavaScript's robust syntax made it learnable through a copy-paste-tweak method. Crockford described it as the rare programming language "most people don't bother to learn before they use."

> You can't do that with any other language, and you shouldn't want to, and you shouldn't do that with this language either. Programming is a serious business, and you should have good knowledge about what you're doing, but most people feel that they ought to be able

> to program in this language without any knowledge at all, and it still works. It's because the language has enormous expressive power, and that's not by accident.[116]

Eich's choice to give JavaScript "enormous expressive power" reflected his decision to design the language for an amateur audience. He sought a "division of labor across the programming pyramid" to "foster greater innovation than alternatives that require all programmers to use the 'real' programming language (Java) … instead of the 'little' scripting language."[117] Lohr's profile of Eich echoed this, declaring JavaScript's purpose "is to give potentially millions of people who are not programmers the ability to modify and embellish Web pages" and characterizing Java as "still a pretty hard-core programming language.[118]" When comparing JavaScript to Sun's Java, Eich himself often characterized the former as Java's "little brother."

Andreessen and Eich displayed remarkable prescience in their broad vision of a ubiquitously networked future, and their vision is revealed in Netscape's mission and in Eich's design choices in JavaScript. Both were regarded for their interests outside of programming, rendered all the more remarkable by their rigorous work schedules to keep pace with Netscape's release schedule. Andreessen saw the web browser as "simple, universal" user interface for a ubiquitously networked future and a way to shape all communications. He envisioned the World Wide Web beyond the PC on "new devices such as smart phones, televisions, and interactive games" and in that eventuality, the operating system would become "plug-ins under the application."[119]

From its founding, Netscape styled itself a communications company, not a software company. Netscape general counsel and senior vice president Roberta Katz described the company as one “that deals with communications” as opposed to “software companies … in the business of ones and zeroes.”

> “What we represent is a fundamental change. It is a new way of thinking about what we do, and it comes from the power of highly scalable networks.”[120]

“[Netscape co-founder Jim] Clark and Andreesen believed that the Web was only the first step toward a networked world. … Netscape's vision did not tie the company to any single product or technology.” Rick Schell, Senior Vice President of Netscape's Client Product Division (responsible for Navigator), described the company's mission statement as “broad,” saying nothing about the internet, and focusing on “connecting people and information on networks,” not restricted “to the Internet or internal networks or anything else” Furthermore, “[t]hat was a very conscious decision.”[121]

Cusumano and Yoffie interviewed Andreessen in 1997 and 1998 and described their meetings as “opportunit[ies] to observe his intellectual eclecticism firsthand.”

> In addition to technology, he liked to talk about business strategy, philosophy, and history, and he enjoyed bringing together the everyday and the arcane. Andreessen's particular talent seemed to lie in making illuminating connections among the discoveries of other thinkers … [which] prove[d] valuable as Andreessen went about the job of articulating Netscape's vision and tying it to specific product-development projects in an industry where the technology and the market were changing all the time.[122]

Crockford, an outspoken critic of Eich, Netscape, and JavaScript, called Eich “brilliant” and credited him for drawing inspiration for JavaScript from Self and Scheme:

> Two really interesting languages, two failed languages in the sense that they got zero market acceptance, but two brilliant languages and two highly influential languages. They’re known not much to the general programming community, but they’re very well

> known to program language designers. It was a surprising choice to take features from those two languages. Scheme was a … [b]rilliant language, probably the best implementation of functions in any language we've ever seen. Self was a brilliant language …[123]

Andreessen at the highest level of Netscape's management and new-hire Eich were both regarded for their broad vision, and their interests and backgrounds led them to be regarded as exceptional people. Andreessen's vision was crucial to Netscape's identity as a company, and when he identified a need for an easily learnable lightweight programming language to run on web pages – something more akin to HTML than Java in complexity – he saw to it that the company developed a solution. Eich was hired explicitly for that purpose and drew from his deep computer science background to develop the language – JavaScript – that would address that need within the context of Netscape's mission. To that end, JavaScript was successful. It remains the only programming language supported natively in every widely available web browser and "[v]irtually every personal computer in the world has at least one JavaScript interpreter installed on it and in active use."[124]

UBIQUITY AND OPEN STANDARDS

This section will describe the evolution of Netscape's relationship to web standards. The design of JavaScript was not based on any existing standards for interactivity on the web. Its proposal to Ecma as a web standard marked a shift in the company's relationship to open standards. Netscape had from the beginning stressed the importance of the browser as the ubiquitous interface for networked society. When there were no browsers competing strongly against Netscape, the company implemented non-standard HTML and other technologies that satisfied their customers, even if they went against the web’s founding mission to be a universal hypertext. However, Netscape’s shift to open standards became necessary due to competitive pressure from Microsoft in the browser market. Netscape had to not only preserve the presence of its products in the market by making sure the technologies they created would be adopted by other software makers. They also had to present themselves as a credible alternative to Microsoft, which had developed a negative reputation for creating proprietary standards.

Between what Crockford described as the dangerous “bad parts” of JavaScript, its “little brother” characterization, its intended audience of amateur programmers, and its early implementation by unseasoned programmers, it quickly developed a negative reputation:

> Its obvious defects, its unfashionable programming model, intentional mispositioning at its introduction, and its ridiculous name caused it to be rejected as unworthy by most knowledgeable programmers.[125]

Netscape’s relationship with the World Wide Web Consortium (W3C), the web’s governing body, had been contentious even before the company was founded, and they were not receptive to the idea of formalizing JavaScript as a standard even as it became a de facto standard through its design and the popularity of the Navigator browser. Although Clark and Andreessen observed that open standards “are not merely desirable” but “essential” in a networked world, Netscape shared with Microsoft a business model founded on ubiquity, and ubiquity based on a *de facto* proprietary standard was better business than being absent from the market.[126]

Netscape's implementation of new code in HTML in the Navigator beta and Andreessen's previous pioneering work supporting images on the web in the Mosaic browser revealed a cavalier attitude toward the W3C's open standards. While disregard for open standards was often poorly perceived among programmers, other Silicon Valley executives expressed their support for Netscape's practices for the more salient purpose of outflanking Microsoft's monopoly on the personal computer platform. As Microsoft entered the browser market with a history of overtly creating proprietary standards, Netscape sought to formalize JavaScript as an open standard as a means of preserving its ubiquity and retaining a competitive advantage in the browser market. The European Computer Manufacturers Association (Ecma), of which Microsoft was a member, accepted Netscape's submission of JavaScript for standardization in November 1996. Both JavaScript and Microsoft's JScript derivative were named originating technologies of EcmaScript, described by Ecma specification 262 (E-262). Eich continues to work on EcmaScript, now in its fifth edition.

Though the concept of "Internet time" has been largely discounted, the concept remains salient in the development of the web browser. The speed with which JavaScript was developed and deployed is evidence of the shifting priorities of browser manufacturers compared to previous software developers. Internet time was not merely a byproduct of aggressive competition between Netscape and Microsoft but also a product of Andreessen's guidance:

> [An] idea which I mentioned earlier is to ship early and often, which is you kick the product out the door as quickly as possible. It doesn't even matter if it's done or doesn't really matter if it does even 20% of what the full expression of it is. Kick it out the door. It may not even work reliably. The purpose is not even so much to start selling it. The purpose is to go out and get feedback on it. The purpose is to go out and get feedback on it so the press-analysts community start writing about it and say you're the leader in the field by virtue of your being there. And also because you're getting feedback from customers, you're getting feedback from people who are going to use it. And they'll come back and they will tell you, often in no uncertain terms, what's wrong with it and what needs to be improved. And that's the most valuable type of input you can really get.[127]

This strategy turned out to be detrimental to the design of JavaScript, and practitioners like Crockford and Eich himself have acknowledged that haste was a source of error: “In a great big hurry, I wrote [JavaScript's] byte-code interpreter. Even at the time I knew I was going to regret some of the things I'd done.”[128] Crockford’s history of JavaScript contextualized its development within the history of computer languages as a “quickfire challenge” and criticized Netscape’s management for allowing it to be released in that condition. By Crockford’s description:

> [T]he biggest influence [for the bad parts], by far, was haste. The language was designed, implemented, and shipped in way too little time. Most languages take years to develop – for example, Smalltalk was eight years from … first prototype to … when it was first made available to the public. That’s a good timeframe for a programming language, because you want to go through it and test it, make sure that it works, and refine it in order to make sure that it’s meeting its goals. JavaScript was prepared in about as many days. It’s amazing that [Eich] could get it done and designed and working in such an incredibly short time; in about two weeks. I challenge any language designer – it’s sort of like a quickfire challenge. That turns out not to be a good way to make software, but that’s how it was done, and we’re now living with the consequences of that. Had Netscape been a better managed company, they might have taken a lot more time, maybe a couple of extra weeks, to clean it up, and we wouldn’t be dealing with the bad parts that we have now. But we have.[129]

JavaScript was built extraordinarily quickly, and Netscape, intent on rapidly deploying their innovations in order to compete against Microsoft, implemented the language with haste. Now the most popular programming language in the world, the decisions that influenced its design and allowed it to be publicly released without sufficient quality control have been writ large.

Computing as a platform

The technologies that came to constitute the web computing platform were shaped by the motivations that companies had for promoting the web as a computing platform. Netscape’s goal of being a ubiquitous interface manifested itself as an investment in cross-platform computing. This goal and investment was supported by other technology firms who saw in Netscape an opportunity to loosen Microsoft's monopoly on the consumer computing industry, key among them Sun Microsystems. Sun's Java programming language was designed to run on multiple platforms but required a "virtual machine" that a user had to install – Netscape agreed to distribute this virtual machine through the browser. With motivation and distribution, companies promoted developing in Java; however, Netscape's in-house JavaScript more easily learned and sufficiently powerful for programmers who sought to create rich applications on the web. The requirements for a platform for rich applications were met by JavaScript, cookies, XmlHttpRequest, and other originally proprietary browser features. This combination of features that became the substrate for Web 2.0 had by 2002 made "the web, together with its associated extension technologies … best described as an infrastructure that web-based applications must rely on for their execution."[130]

Netscape's naked ambition was to render "operating systems [as] plug-ins under the [browser]," and Sun saw in Navigator a medium to deploy Java and break Microsoft's lucrative and seemingly natural monopoly on the operating system and computing platforms based on Intel's x86 architecture, including the vast majority of consumer personal computers and a growing number of high-end enterprise machines.[131] Economic analysis of U.S. v. Microsoft cited this as the primary motivation for Microsoft's serious entrance into the browser market, revealing internal documents that showed "Microsoft recognized that the dominant position of its Windows operating system could be threatened by an Internet browser that was capable of supporting applications independent of the operating system."[132]

On 7 December 1995, Microsoft CEO Bill Gates called a company-wide meeting to outline a new internet strategy based on the browser, abruptly abandoning investment in internet ventures like media portal MSN and interactive television. Microsoft managers convinced Gates of the threat Navigator posed to their natural monopoly on the operating system because it eroded the "applications barrier to entry." Through a combination of network effects and vendor lock-in, Microsoft operating systems had come to amass an install base estimated at over 90% of the world's computers, and the library of 70,000 programs that existed for Windows dwarfed the library of 12,000 that existed for its closest competitor, Apple Mac OS. Consumers' choice of "Wintel" (Windows OS + Intel x86 hardware) platforms was influenced by this software library, and independent software vendors tended to develop for the platform with the largest installed base. Further enforcing the lock-in, Windows' proprietary application programming interfaces (APIs) were also designed to make programs designed for Windows difficult for independent software vendors to re-make for other platforms.[133]

As computers became more widely adopted in the 1990s, Microsoft became one of the largest companies in the world and founder and Gates became the world's richest man. However, during this time he developed a reputation as a strongman for his business strategy, which often involved eliminating Microsoft's competition. By 1995, other technology executives, including Clark, had grown weary of Microsoft's strategy – Clark himself had been muscled out of the high-end market he once dominated as the founder and chairman of Silicon Graphics. Netscape programmer Lou Montulli described Microsoft as "the antithesis of the Internet, or at least the way the Internet was." Microsoft's reputation had led to "thousands of disaffected software developers and Microsoft competitors looking for alternatives to Microsoft," and "these companies, often called the ABM crowd, ‘Anybody But Microsoft,’ were a natural constituency to align with Netscape.”[134] McNealy shared Clark’s weariness and Java was promoted heavily by technology firms including both “customers and competitors of Microsoft:” IBM, Compaq, Oracle, and Cisco “were willing to invest $100 million collectively to create the Java Fund to support start-ups who would use Java to challenge Microsoft’s hegemony.”[135]

Netscape’s decision to design Navigator for multiple platforms, effectively as a superstructure overlaying all operating systems, was meant to turn Microsoft's installed base into a liability, and Cusumano and Yoffie's analysis of the company identified “the company's most significant source of leverage against Microsoft came from its investment in cross-platform design.”[136] Cross-platform design had failed in ventures by firms as large as IBM and Apple because it required developers to “not utilize any interfaces or programming ‘tricks’” that would allow them to “write code that runs faster or handles graphics and memory better than code that uses the ‘lowest-common-denominator’ interfaces.”[137] Different platforms supported different file formats for typography and images, and programmers resisted designing interfaces with these limits.[138] While at NCSA, “small teams worked in parallel, competing with each other to finish browser versions for Windows 3.1, Macintosh, and some of the UNIX operating systems.”[139] Aleks Totic and other engineers from the Mosaic team who joined Netscape developed Navigator 1.0 to run on these same “three platforms successfully, and it quickly became the browser of choice for nearly all desktop computers.”[140] By 1996, Netscape considered developing their Communicator software suite wholly in Java, making it natively cross-platform.[141]

The licensing of Java and development of JavaScript gave Netscape a pair of technologies with which content authors and coders – increasingly representing commercial interests – could program interactive and animated presentations on that superstructure instead of the platform underneath. These set new precedents not only for interactivity but the role of plug-ins and programming languages for the web. Plug-ins, third-party desktop applications running inside the browser, including Flash and Acrobat (PDF) Reader gave content authors and web surfers access to rich media presentations and PostScript-quality design.

Netscape and Microsoft added other proprietary features to be used natively to the browser like cookies and XmlHttpRequest. Cookies, which allow websites to store information on a user's hard drive, were originally invented by Lou Montulli at Netscape for use in an e-commerce shopping cart.[142] In the development of the web platform, they effectively gave the browser a kind of random-access memory and allowed web applications "to keep track of application execution," "to distinguish among application instances," and "to distinguish among different users."[143] XmlHttpRequest was developed by Microsoft in 2000 to allow a JavaScript program running in a browser to call a server for new content and present that content to a user. XmlHttpRequest, now a W3C draft, allowed applications written in JavaScript to send and receive content from a server without reloading the page or going to a new page. Effectively, it allowed users to seamlessly interact with servers – an important aspect of Web 2.0 applications that make it "make it possible to build web sites that behave much like desktop applications, for example, by allowing web pages to update one user interface element at a time, rather than requiring the entire page to be updated each time something changes."[144]

This feature of the web platform, however, exists at odds with the convention of the URL – circumventing the HTTP request made web applications appear seamless and moved the web "in directions unforeseen by its designers, with web sites behaving more like multimedia presentations rather than conventional pages."[145] Maps could be scrolled infinitely, heavy audio and video could be played with smaller buffers, web mail applications could check for new messages without affecting a user's activity, and long lists of content could be loaded piecemeal into a single page. However, while these are presented to the user, the browser can not store – and allow a user to manage – information about the parts of the web they have seen. The invention and rapid adoption of JavaScript "made it possible to build animated, interactive content more easily" and allowed navigation "no longer based solely on links" as "communication between the browser and the server became increasingly advanced." By allowing coders to circumvent the HTTP request for navigating the web, JavaScript and other technologies were able to effectively provide programmers with a computing platform within the browser. As the browser's interface remains designed to navigate the web as a hypertext rather than a computing platform, new networked computing platforms such as smartphone operating systems have emerged to address the schism in the web's *raison d'etre* opened with the invention of JavaScript.

CONCLUSION

> Given the process that created JavaScript and made it a de facto standard, we deserve something far worse."
>
> —Crockford, quoted by Eich.[146]

Constituent technologies of the web platform are now conventions of the web and almost all *de jure* standards ratified by the W3C (with the exception of JavaScript). They have come to form the foundation of products by Web 2.0 firms like Google and changed how software is created and distributed. However, the browser designed to navigate the web as a hypertext remains insufficient for navigating the web as a computing platform. Understanding the goals and decisions that informed the design of these constituent technologies is key to understanding how these applications are structured and limited by the browser. The so-called 'death of the web' and the rise of purpose-built applications are responses to the development of the web as a computing platform though its interface conventions largely remained true to its roots as a hypertext system.

JavaScript is an anomaly among the stack of languages that comprise the World Wide Web. Unlike W3C standards like HTML and Cascading Style Sheets, which were developed and standardized by the IETF and W3C over several years with public commenting processes, JavaScript was created by one man in one company in ten days. It was successful because it addressed two needs: it was designed to be quickly learned and for more than a year – a year distinguished by the introduction of the web to the general public – it was only one of two ways to facilitate interaction and animation on the new medium of the web page. It was so successful that when Microsoft attempted to address this in Internet Explorer without licensing JavaScript outright, they did not set out to create a new standard but instead reverse-engineered the language to create JScript – effectively conceding to Netscape the status of JavaScript as a de facto standard. JavaScript was meant to be easily adopted, and its reputation for misuse by amateur programmers is arguably inherent to its design.

However robust JavaScript has proven as a programming language, it also represented a young, ambitious company's failure – through philosophy, background, and market influences – to properly assess a flawed product before releasing it to the public. The fact that that company (Netscape) by virtue of releasing that product (Navigator 2.0 with support for Java and JavaScript) entered an adjacent market (operating systems) dominated by one of the largest companies in the world (Microsoft) and inflamed them to start a format war (the browser wars) prevented JavaScript from transcending the flaws of its initial design. The format war that escalated following JavaScript's release is evidence of the impact of the design of the web browser on the shape of the information economy; the enduring popularity of JavaScript is at once evidence of the lasting effects of the browser wars and the quality of the ideas upon which it was based.

JavaScript became one of a series of foundation technologies that allowed the web to function as a computing platform distinct from the operating system and independent of the hardware on which it runs. This was the outcome Netscape had sought in their development of the browser, and while the company lost the browser wars, the browser features it invented and implemented were widely adopted. After the release of Navigator 2.0 with support for JavaScript, the "Web started moving in directions unforeseen by its designers, with web sites behaving more like multimedia presentations rather than conventional pages,"[147] and features like cookies and XmlHttpRequest allowed the browser to be a platform for rich software applications. In recent years, JavaScript has been viewed more favorably as an instrumental part of Ajax, a suite of languages and features that has defined interactive websites since 2004. These applications came to change the software industry by making the web "the *de facto* deployment environment for new software systems and applications."[148]

CHAPTER 5 CONCLUSION

Concerned by present debates and misunderstandings about how people use the web, I entered my research with the intention of studying how disciplines ranging from computer science to the humanities were affected by the interface of the web browser. While I approached the browser as the product of human endeavors, my initial definition of "browser makers" was limited to employees of firms Netscape and Microsoft or participants in open source projects like Mozilla and WebKit. However, the software engineers who built browsers and executives who oversaw them were only two groups among many who structured the web. In two historical case studies on uses of HTML for graphic design and the development of interactivity on the web, synthesis of archived emails from members of the IETF, lectures and papers by programmers and executives, socio-technical studies of the web design profession, and legal analysis and journalistic accounts of the browser wars revealed how groups such as graphic designers and government regulators were enrolled in the development of the web.

Applying the social construction of technology (SCOT) theoretical framework to this history organizes the myriad debates about the evolution of the web into a few social groups guided by a contrasting pair of motivations: the potential of the web to be a universal hypertext system and the vision of Netscape's browser as the ubiquitous interface of a networked society. In analyzing the case of proprietary HTML elements and their role in transforming the web into a graphic design medium, the elements Netscape developed became, through the work of graphic designers, a foundation employed by commercial interests on the web. However, these elements also jeopardized the web's potential to be the universal hypertext system imagined by its inventor. In the construction of interactivity on the web, the design of JavaScript was guided by its inventor's academic background, Netscape's guidance to produce a "glue language," and corporate maneuvering around Microsoft's natural monopoly of the personal computing platform through the 1990s.

While these two cases are monumental in the history of the web, they are early examples that offer a direction for continuing research on the form of the web browser in the SCOT framework. The following sections describe three possible trajectories for research following these case studies.

Beyond the SCOT framework

A continuation of this inquiry could take the form of a re-examination of the SCOT model for software and networked technologies, because the browser is in some ways an odd fit for the SCOT framework. While the SCOT model assails the idea of lonely inventors and "eureka" moments as bases for invention, the World Wide Web and features like JavaScript have identifiable inventors – Berners-Lee and Eich. While each of them situates their work in a lineage of preceding ideas, like Xanadu and obscure programming languages like Self and Scheme, and social milieus, like CERN's proposal process and Netscape's executive mandate, each of them are factually the inventors of the World Wide Web and JavaScript.

Landmark case studies that serve as examples of the SCOT framework cover technologies like the bicycle and Bakelite which are built from finite raw materials and are limited in their production and distribution compared to software. The browser was distributed freely and at nominal marginal cost to Netscape and Microsoft. Successive versions of browsers were designed to replace existing installations, and new browsers could be shipped to all of their users in a matter of minutes.

Because the web's protocols are governed by an open license by CERN, anyone can produce a browser without paying royalties to the web's inventors. Further, because the web's power is partly due to the interoperability of its pages, extending the functionality of the browser can threaten its utility.

THE WEB AND ITS STANDARDS, AFTER 1995

While the developments of proprietary HTML and JavaScript that led to the commercialization of the web and its reformulation as a computing platform were milestones that define the start of the modern web, they are only two of numerous features and protocols developed by browser makers to advance the web's capabilities. These features and protocols, culminate in a contemporary, stable definition of the browser. This definition, contrasted with the execution of the web as platform manifest in popular web applications, would more directly address the browser's deficiencies compared to newer smartphone-based platforms.

Cookies, a string of text that a browser can save on a user's hard drive, were invented by Lou Montulli for use in an e-commerce shopping cart application developed by Netscape. They became drafted as a standard by the IETF for state management on websites through RFCs 2109 and 2965 and are instrumental in the web's computing platform for effectively allowing web-based applications to have random-access memory. The "persistent" variation of cookies has also been a foundational technology in behavioral advertising[149], a source of billions of dollars in revenue for large internet companies and proposed regulation by the U.S. Federal Trade Commission.[150]

Mozilla's addition of tabbed browsing and a search form to the interface of the Phoenix/Firefox browser in 2001 was a formative step towards the browser's modern form. Tabbed browsing allowed users to multitask on the web within a single window, staking out more territory previously occupied by the operating system. By a business arrangement with Google, the search engine provides a vast amount of the Mozilla Foundation's operating budget in exchange for being Firefox's default search engine.[151]

Other features that would merit similar analyses include the HTTPS protocol for secure transactions, now used uniformly by websites that process sensitive information; browser caching, a method of saving files on a web surfer's computer for improved performance that has ramifications on privacy; and web-native multimedia like audio and video, which are currently debated among browser makers who have not settled on uniform file formats.

THE WORLD WIDE WEB OF EVERYONE

The web has come to permeate so much of everyday life that the number of social groups presently shaping its evolution seems innumerable. In addition to being a testament to the effects the web has had throughout society, it suggests a continuation of this research on the roles played by government regulators, open source software advocates, and standards-compliant designers in shaping the browser and the web.

This nascent computing platform and Netscape's IPO spurred Microsoft to pour resources into developing their own browser, Internet Explorer, starting the "browser wars" that were the foundation for the concept of "internet time." Microsoft's realization in 1995 that the web not only was a potentially enormous business but a threat to its own lucrative monopoly on personal computing led the company to pursue anticompetitive business practices against Netscape, culminating in the landmark *U.S. v. Microsoft* antitrust case and similar complaints in the European Union. Microsoft's anticompetitive practices – preventing customers of their Windows operating system like Compaq and Gateway from shipping computers with Netscape installed – led to the landmark *U.S. v. Microsoft* antitrust case, a case that had "important implications for how governments regulate information technologies and the … Internet explosion."[152]

As Netscape's market share withered from Microsoft's attack, Brendan Eich was among those leading the push for the company to release their browser's source code to the public through the Mozilla project. The project made the world's second most popular browser and a first-class software application open source, and as of April 2011, the Firefox browser that project begat remains an open source project with 22% share of the browser market worldwide.[153] Combined with the market shares of Google Chrome and Apple Safari (based on the open source WebKit rendering engine), 41% of web surfers – hundreds of millions of people – regularly use an open source software product.[154]

Advocates for standards emerged from the community of web designers frustrated by the redundant work necessitated by the inconsistent implementations of HTML between browsers. Groups like the Web Standards Project (WaSP) formed and successfully lobbied makers of browsers, authoring tools, and web designers to adhere to web standards. Groundbreaking support for web standards like Cascading Style Sheets – a language for web design – was implemented in Internet Explorer version 6 for Windows and version 5 for the Macintosh operating system in 2001. Among designers, developing web sites with tables and one-pixel images is considered passé and wasteful of bandwidth resources. Jeffrey Zeldman, WaSP co-founder and creative director of the web design firm Happy Cog[155], wrote "the web design movement's foundational text"[156] *Designing With Web Standards* in 2003 to address web designers' history of "solv[ing] today's problems at tomorrow's expense."[157] The book is now in its third edition and Zeldman's own press, A Book Apart, now publishes titles like *HTML5 For Web Designers* that introduce designers to emerging web standards and provide guidance for their use.

Makers of modern browsers like Firefox and Opera tout their adherence to standards, and the development of HTML5 by browser makers through a splinter group of the W3C reflected browser makers' acknowledgement of standards' importance while evoking precedents for ratifying *de facto* standards into the HTML specification. In RFC 1942 that specified the *table* element, the "cellspacing" and "cellpadding" attributes that Netscape added to its implementation of the *table* element were officially drafted into the specification, "intended for backwards compatibility with deployed user agents."[158] After Andreessen proposed the *img* element to NCSA Mosaic, the IETF added *img* to the HTML 1.0 specification. The W3C specification for HTML 4 edited by Raggett, Arnaud Le Hors, and Ian Jacobs and recommended in December 1999 included support for the *frameset* tag and its child elements developed by Netscape. One of Ian Hickson's guiding philosophies as editor of HTML5 was a similar "pave the cowpaths" approach to setting standards – recognizing the widely used practices that became *de facto* standards as *de jure* standards of the web.[159]

Before Netscape, HTML was a markup language different browsers rendered in different ways; Mosaic/Netscape's decision to add formatting elements to HTML extended control over the look of a page to the people who coded them, and the mating of design principles with coding expertise gave rise to the web design profession and commercial interests.

Tools like WaSP's ACID tests provide a reference implementation for modern HTML and CSS specifications against which browser makers measure their products, complementing the written specifications of web standards including HTML5. The web's syntactic interoperability is regarded positively for making the experience of using the web more consistent between browsers and platforms. While the rise of new smartphone-based platforms like iOS and Android have spelled the "end of the web" for some pundits, standards-compliant browsers and design practices have allowed these platforms to access the web's universal hypertext system.

IN SUM

The growth of websites and companies born on the web remains structured and bounded by the designs of the browser and the web's languages. Lines of research on the web's economic impact in various industries, cognitive impact on readers of hypertexts, and cultural impact on institutions and individuals benefit from an understanding of the form of the browser and the work of its various makers. By shaping the software people use in order to use the web, browser makers define the web itself, providing the basic interface and set of technologies for all e-commerce shoppers and retailers, people connecting on social networking services, researchers querying search engines, and viewers of hypertext art.

On the web as in other networked systems, open standards are crucial to the system's survival. During the early modern web, characterized by Netscape's dominance and the subsequent browser wars, Netscape's and Microsoft's release schedules lapped the IETF's "expeditious" ten-month standards-drafting process. Netscape's implementation of HTML's *table* element, invention of JavaScript, and standardization of JavaScript as Ecma standard E-262 reflected browser makers' evolving relationship with open *de jure* standards as the web evolved. Microsoft, by mimicking Netscape through the design of Internet Explorer's interface and the creation of the JScript language, cemented these as *de facto* web standards. Netscape's addition of interactivity to the web through JavaScript and the subsequent browser wars between Netscape and Microsoft led to even more proprietary HTML elements like *layer, spacer*, and *marquee* that strained the relationship between browser makers and standards writers and simultaneously made the web increasingly popular while jeopardizing its potential for universal interoperability.

David Siegel, the designer who popularized the technique of using HTML tables for graphic design, captured this paradox when he observed that when Marc Andreessen added inline images to web pages, "the Web was both born and destroyed at that moment."[160] By defying standards, browser makers transformed an exciting academic project into a global mundanity.

GLOSSARY

The following definitions of key names and technical terms from this thesis complement the glossary appended to Berners-Lee's *Weaving the Web*. For additional terms, see http://www.w3.org/People/Berners-Lee/Weaving/glossary.html.

Andreessen, Marc

Co-writer of Mosaic browser, co-founder of Netscape Communications Corporation. See http://about.ning.com/management/.

Berners-Lee, Tim

Inventor of the World Wide Web: he wrote the first web browser (WorldWideWeb), created HyperText Markup Language (HTML), and wrote the HyperText Transfer Protocol (HTTP). See http://www.w3.org/People/Berners-Lee/.

Clark, Jim

Co-founder of Netscape Communications Corporation. Founder of Silicon Graphics, Inc., manufacturers of high-end computer workstations used for graphics, visual effects, and 3-D editing. See Lewis, *The New New Thing*.

CSS

Cascading Style Sheets: a web standard for formatting web pages. A style sheet is a separate document from a web page, allowing the content of a page to still be usable and semantically rich in cases where formatting is irrelevant, as in machine-to-machine communications.

Eich, Brendan

Computer programmer employed by Netscape, previously employed by Silicon Graphics, Inc. Inventor of JavaScript. See Lohr; also: http://classic-web.archive.org/web/20000815055653/people.netscape.com/brendan/.

File formats

A method of encoding a computer file. In order for text, graphics, audio, video, etc. to be transferred between computers, it must be placed in a file – like a container – of a particular format – like a cipher – that both computers can read and write. Some examples include JPG (Joint Photographic Experts Group File Interchange Format), GIF (Graphics Interchange Format), and PNG (Portable Network Graphics) for images.

HTML

HyperText Markup Language, see Berners-Lee.

Internet Engineering Task Force (IETF)

An international body charged with overseeing the development and maintenance of the internet's architecture. See http://www.ietf.org/about/.

JavaScript

Programming language invented by Brendan Eich while employed at Netscape. Originally called Mocha, then LiveScript, it was written in less than two weeks – an extraordinarily short period of time for a programming language. See Seibel, Eich (all), and Crockford (all).

Mosaic

The first popular web browser. Developed by Marc Andreessen and Eric Bina, student employees of the National Center for Supercomputer Applications (NCSA) at the University of Illinois at Urbana-Champaign.

Netscape

Netscape Communications Corporation: Company that developed and shipped Netscape Navigator, the first popular modern web browser.

plug-in

Applications that run within web browsers in order to provide extra functionality. Some popular examples include Adobe Flash (used for audio/video content and games)

tables

In HTML, a set of elements - *table*, *thead*, *tr*, *th*, *td* - that were developed to allow authors of websites to add data tables to documents. Because of their flexibility, they were used with images for graphic design.

view source

A feature in web browsers that allows users to view the source code of any page on the web.

web standards

A series of languages including HTML and CSS that are recommended by the World Wide Web Consortium for usage by browser makers and page authors in order for any website to be usable in any browser. See http://www.webstandards.org/about/mission/ and Zeldman.

World Wide Web Consortium (W3C)

An international organization of corporations and academic institutions that develops web standards. See http://www.w3.org/Consortium/.

NOTES

[1] See Human Rights Watch, "How Censorship Works in China: A Brief Overview," August 2006. Smith, Josh. "Egypt Shutters Internet in Unprecedented Move, Analysts Say." National Journal. 28 January 2011. Rhoads, Christopher and Loretta Chao. "Iran's Web Spying Aided By Western Technology." Wall Street Journal. 22 June 2009.

[2] Initial public offering.

[3] Rusli and Kopytoff. "Investing Like It's 1999."

[4] Ibid.

[5] GOOG stock quote, accessed 2 April 2011.

[6] Sage. "Amazon's projected cost of growth hits shares hard."

[7] Herman. "A process for creating the business case for user experience projects."

[8] Wright. "New Search Technologies Mine The Web More Deeply."

[9] Manjoo. "7 Ways Larry Page Is Defining Google's Future."

[10] Gruber. "Apple, Adobe, Flash."

[11] Cusumano and Yoffie, *Competing on Internet Time* 25.

[12] Berners-Lee, *Weaving The Web* 157.

[13] Libicki et al 25.

[14] Berners-Lee, *Weaving The Web* 158.

[15] Anderson and Wolff, "The Web is Dead, Long Live the Internet."

[16] Ibid.

[17] Agarwal; Hewitt, specifically status updates 13090747143, 13032980959, 13090363860, 13090720669, 13091031807, 13091325835, 13094164197, 13094261355, 13094326988, 13094617071, 13095015896, 13095170856, 13095254263, 13095304769, 13095460167, 13097165783, and 13097223950.

[18] HTML5. See Keith.

[19] Specifically CSS transformations by Apple Safari and geolocation by Google Chrome. See Katz.

[20] Katz, "The Web Doesn't Suck. Browsers Are Innovating"; Blizzard, "Innovation in Browsers."

[21] Gates, "Microsoft's Tradition of Innovation."

[22] Fisher and Rubenfeld 34-35.

[23] Fisher and Rubenfeld 9-12.

[24] Jobs, "Thoughts on Flash."

[25] Oudshoorn and Pinch, *How Users Matter* 24.

[26] Oudshoorn and Pinch, *How Users Matter* 3.

[27] Bijker and Pinch, "Social Construction of Facts and Artefacts" 421.

[28] Bijker and Pinch, "Social Construction of Facts and Artefacts" 416.

[29] Ibid.

[30] Ibid.

[31] Bijker and Pinch, "Social Construction of Facts and Artefacts" 427.

[32] Bijker, "The Social Construction of Bakelite: Towards a Theory of Invention" 168.

[33] Bijker, "How Is Technology Made" 69.

[34] Bijker and Pinch, "Social Construction of Facts and Artefacts" 414.

[35] Berners-Lee, "HyperText and CERN."

[36] Berners-Lee, *Weaving the Web* 50.

[37] Berners-Lee, *Weaving the Web* 82.

[38] Berners-Lee, *Weaving the Web* 78-84.

[39] Cusumano and Yoffie, *Competing on Internet Time* 22.

[40] Cusumano and Yoffie, *Competing on Internet Time* 73-74; Lewis 73-75.

[41] Berners-Lee, "HyperText and CERN."

[42] Ibid.

[43] Ibid.

[44] Ibid.

[45] Berners-Lee, "HyperText and CERN." See also Faraj, Kwon and Watts.

[46] Berners-Lee, "HyperText and CERN."

[47] Berners-Lee, *Weaving the Web* 68.

[48] Berners-Lee, *Weaving the Web* 69.

[49] Ibid.

[50] Libicki et al 5.

[51] Ibid.

[52] Berners-Lee, *Weaving the Web* 74.

[53] Web History Project: Andreessen, Marc. "proposed new tag: IMG." 25 February 1993. http://1997.webhistory.org/www.lists/www-talk.1993q1/0182.html.

[54] Web History Project: Berners-Lee, Tim. "Re: proposed new tag: IMG." 26 February 1993. http://1997.webhistory.org/www.lists/www-talk.1993q1/0191.html

[55] Web History Project: Andreessen, Marc. "NCSA Mosaic for X 0.10 released." 14 March 1993. http://1997.webhistory.org/www.lists/www-talk.1993q1/0262.html; "NCSA Mosaic 1.0 released." 21 April 1993. http://1997.webhistory.org/www.lists/www-talk.1993q2/0128.html.

[56] Berners-Lee, *Weaving the Web* 83.

[57] Web History Project: Andreessen, Marc. "Mosaic Netscape is out the door…" 13 October 1994. http://1997.webhistory.org/www.lists/www-talk.1994q4/0187.html

[58] Web History Project: *www-talk*, http://1997.webhistory.org/www.lists/www-html.1994q4/subject.html#start.

[59] Web History Project: Northrup, Dylan. "Re: Netscape & New HTML." 21 Oct 1994. http://1997.webhistory.org/www.lists/www-html.1994q4/0026.html.

[60] Web History Project: Behlendorf, Brian. "Re: Netscape & New HTML." 21 Oct 1994. http://1997.webhistory.org/www.lists/www-html.1994q4/0028.html.

[61] Ibid.

[62] Web History Project: Northrup, Dylan. "Re: Netscape & New HTML." 21 Oct 1994. http://1997.webhistory.org/www.lists/www-html.1994q4/0026.html.

[63] Web History Project: Behlendorf, Brian. "Re: Netscape & New HTML." 21 Oct 1994. http://1997.webhistory.org/www.lists/www-html.1994q4/0024.html.

[64] Berners-Lee, "The World Wide Web - Past, Present and Future."

[65] Libicki et al 5.

[66] Libicki et al 6.

[67] Russell.

[68] Kotamraju, “Keeping Up: Web Design Skill and the Reinvented Worker” 15.

[69] Ibid.

[70] Kotamraju, “Keeping Up: Web Design Skill and the Reinvented Worker” 16.

[71] Kotamraju, “Keeping Up: Web Design Skill and the Reinvented Worker” 12.

[72] Kotamraju, “Keeping Up: Web Design Skill and the Reinvented Worker” 11.

[73] Quinn 51.

[74] Ibid.

[75] Siegel.

[76] Müller-Brockmann 13.

[77] Jobs, Steve, quoted by Wolf, “Steve Jobs: The Next Insanely Great Thing.”

[78] Ibid.

[79] Ibid.

[80] Raggett.

[81] Ibid.

[82] DiNucci, Giudice and Stiles 24.

[83] Siegel.

[84] Ibid.

[85] Ibid.

[86] Raggett.

[87] Siegel.

[88] Siegel.

[89] Web History Project: Lilley, Chris. “About wanting control over the look of a page.” 31 May 1994. http://1997.webhistory.org/www.lists/www-html.1994q2/0010.html.

[90] Ibid.

[91] Siegel.

[92] Berners-Lee, “The World Wide Web - Past, Present and Future.”

[93] See Microsoft Corporation, “A history of Windows.” http://windows.microsoft.com/en-us/windows/history and Fortune 500 (industrial) 1991-1997, overall 1998-present.

[94] Cusumano and Yoffie, *Competing on Internet Time* 133.

[95] Bank.

[96] McNealy, Scott, quoted by Bank.

[97] Lewis 58.

[98] Ibid.

[99] Andreessen, Marc, quoted by Cusumano and Yoffie, *Competing On Internet Time* 24.

[100] Wang, Wu and Lin.

[101] Mayo and Newcomb.

[102] Libicki et al 22.

[103] Ibid.

[104] Pattinson and Brown 33

[105] Ibid.

[106] Libicki et al 6.

[107] Berners-Lee, *Weaving the Web* 157.

[108] Web History Project: Andreessen, Marc. "Indented <MENU>s." 13 February 1994. http://1997.webhistory.org/www.lists/www-talk.1994q1/0648.html.

[109] Lie.

[110] Hamilton.

[111] Ibid.

[112] O'Reilly.

[113] Ibid.

[114] Cusumano and Yoffie, "What Netscape Learned From Cross-Platform Design" 74.

[115] Most programming languages require the programmer to "compile" the program before it can be run. It's an extra step in the process of writing a computer program that, depending on the size of the program, can add a lot of time even when making minor changes to a program. HTML, JavaScript, and other scripting/programming languages native to the web do not need to be compiled, so someone learning these languages can see how their changes are rendered very quickly.

[116] Crockford, "Crockford on JavaScript -- Chapter 2: And Then There Was JavaScript."

[117] Hamilton.

[118] Lohr.

[119] Cusumano and Yoffie, *Competing on Internet Time* 23.

[120] Katz, Roberta, quoted by Cusumano and Yoffie, *Competing on Internet Time* 23.

[121] Schell, Rick, quoted by Cusumano and Yoffie, *Competing on Internet Time* 27.

[122] Cusumano and Yoffie, *Competing on Internet Time* 20.

[123] Crockford, "Crockford on JavaScript -- Chapter 2: And Then There Was JavaScript."

[124] Crockford, “JavaScript: The World's Most Misunderstood Programming Language.”

[125] Crockford, “The World's Most Misunderstood Programming Language Has Become the World's Most Popular Programming Language.”

[126] Cusumano and Yoffie, *Competing On Internet Time* 25.

[127] Andreessen, speech at MIT, 14 November 1996, from notes provided by Michael A. Cusumano.

[128] Eich, Brendan, interviewed by Seibel, *Coders at Work* 150.

[129] Crockford, “Crockford on JavaScript -- Chapter 2: And Then There Was JavaScript.”

[130] Zhao, Kearney and Gioiosa.

[131] Cusumano and Yoffie, *Competing on Internet Time*; Bank.

[132] Fisher and Rubinfeld 14.

[133] Fisher and Rubinfeld 8-12.

[134] Cusumano and Yoffie, *Competing on Internet Time* 133.

[135] Cusumano and Yoffie, *Competing on Internet Time* 135.

[136] Cusumano and Yoffie, *Competing on Internet Time* 157.

[137] Cusumano and Yoffie, *Competing on Internet Time* 159.

[138] Cusumano and Yoffie, *Competing on Internet Time* 159-160.

[139] Cusumano and Yoffie, *Competing on Internet Time* 158.

[140] Ibid.

[141] Cusumano and Yoffie, *Competing on Internet Time* 165, 174.

[142] Montulli and Kristol.

[143] Zhao, Kearney and Gioiosa.

[144] Taivalsaari et al.

[145] Ibid.

[146] Eich, "JavaScript, A Brief History."

[147] Ibid.

[148] Ibid.

[149] Behavioral advertising utilizes a user's browsing history to determine their interests and demographics and display advertisements that are likely relevant. Behavioral advertising companies often acquire a user's browsing history through a cookie stored on a user's local computer.

[150] Federal Trade Commission.

[151] Mozilla Corporation.

[152] Hahn and Litan vii.

[153] Bright.

[154] Bright.

[155] A prominent web design firm. Of interest, they are the designers of georgetown.edu.

[156] Happy Cog.

[157] Zeldman.

[158] Raggett.

[159] Keith 10.

[160] Siegel.

WORKS CITED

"About Jeffrey Zeldman." *Happy Cog*. Web. <http://happycog.com/about/zeldman/>.

Agarwal, Sachin. "The Web Sucks. Browsers Need to Innovate - Sachin's Posterous." *Sachin's Posterous - My Life: London, LA, Stanford, Phi Psi, Apple, New York City, Posterous, San Francisco, Kate.* 29 Apr. 2010. Web. 28 Apr. 2011. <http://sachin.posterous.com/the-web-sucks>.

Anderson, Chris, and Michael Wolff. "The Web Is Dead. Long Live the Internet." *Wired* Sept. 2010. Web.

Andreessen, Marc. "Marc Andreessen at MIT." Massachusetts, Cambridge. 14 Nov. 1996. Speech.

Transcribed and annotated by Michael A. Cusumano.

Bank, David. "The Java Saga." *Wired* Dec. 1995. Web.

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. "HyperText and CERN." *World Wide Web Consortium (W3C)*. Mar. 1989. Web. <http://www.w3.org/Administration/HTandCERN.txt>.

Berners-Lee, Tim. "The World Wide Web: Past, Present and Future." *Journal of Digital Information*. 17 July 1996. Web.

Berners-Lee, Tim. "WorldWideWeb: Proposal for a HyperText Project." *World Wide Web Consortium (W3C)*. 12 Nov. 1990. Web. <http://www.w3.org/Proposal.html>.

Bijker, Wiebe E. "How Is Technology Made." *Cambridge Journal of Economics* 34.1 (2010): 63-76. Print.

Bijker, Wiebe E. "The Social Construction of Bakelite: Towards a Theory of Invention." *The Social Construction of Technological Systems: New Directions in the Sociology and History of Technology*. Ed. Wiebe E. Bijker, Thomas Hughes, and Trevor Pinch. Cambridge: MIT, 1989. 159-87. Print.

Blizzard, Christopher. "Innovation in Browsers." Web log post. *Christopher Blizzard*. 29 Apr. 2010. Web. <http://www.0xdeadbeef.com/weblog/2010/04/innovation-in-browsers/>.

Bright, Peter. "Web Browser Market Share: Modern Browser Edition." *Ars Technica*. 2 Apr. 2011. Web. <http://arstechnica.com/web/news/2011/04/web-browser-market-share-modern-browser-edition.ars>.

Bright, Peter. "Web Browser Market Share: Modern Browser Edition." Web log post. *Ars Technica*. Web. <http://arstechnica.com/web/news/2011/04/web-browser-market-share-modern-browser-edition.ars>.

Cothey, Vivian. "A Longitudinal Study of World Wide Web Users' Information-searching Behavior." *Journal of the American Society for Information Science and Technology* 53.2 (2002): 67-78. Print.

Crockford, Douglas, and Brendan Eich. "Brendan Eich - ECMA Harmony and the Future of JavaScript." Lecture. *Yahoo! Developer Network*. 29 Oct. 2009. Web. 19 Dec. 2010. <http://developer.yahoo.com/yui/theater/video.php?v=eich-yuiconf2009-harmony>.

Crockford, Douglas. "Crockford on JavaScript -- Chapter 2: And Then There Was JavaScript." *Yahoo! Developer Network*. 7 Feb. 2010. Web. 19 Dec. 2010. <http://developer.yahoo.com/yui/theater/video.php?v=crockonjs-2>.

Crockford, Douglas. "JavaScript: The World's Most Misunderstood Programming Language." *Douglas Crockford's Javascript*. Web. 05 Dec. 2010. <http://javascript.crockford.com/javascript.html>.

Crockford, Douglas. "JavaScript: The World's Most Misunderstood Programming Language Has Become the World's Most Popular Programming Language." *Douglas Crockford's Javascript*. 3 Mar. 2008. Web. <http://javascript.crockford.com/popular.html>.

Cusumano, Michael A., and David B. Yoffie. *Competing on Internet Time: Lessons from Netscape and Its Battle with Microsoft*. New York, NY: Free, 1998. Print.

Cusumano, Michael A., and David B. Yoffie. "What Netscape Learned From Cross-Platform Design." *Communications of the ACM* 42.10 (1999): 72-78. Print.

DiMaggio, Paul, Eszter Hargittai, W. Russell Neuman, and John P. Robinson. "Social Implications of the Internet." *Annual Review of Sociology* 27.1 (2001): 307-36. Print.

DiNucci, Darcy, Maria Giudice, and Lynne Stiles. *Elements of Web Design*. Berkeley, CA: Peachpit, 1998. Print.

EcmaScript Language Specification. Aug. 1998. 2nd Edition. Geneva.

Eich, Brendan. "Brendan's Roadmap Updates: Popularity." *MozillaZine Weblogs*. Web. 05 Dec. 2010. <http://weblogs.mozillazine.org/roadmap/archives/2008/04/popularity.html>.

Eich, Brendan. "JavaScript, a Brief History." JS-Montreal. Montreal. 3 Apr. 2010. Lecture.

Faraj, Samer, Dowan Kwon, and Stephanie Watts. "Contested Artifact: Technology Sensemaking, Actor Networks, and the Shaping of the Web Browser." *Information Technology & People* 17.2 (2004): 186-209. Print.

Federal Trade Commission. *Protecting Consumer Privacy in an Era of Rapid Change: A Proposed Framework for Businesses and Policymakers*. Rep. 2010. Print.

Fisher, Franklin M., and Daniel L. Rubinfeld. "United States V. Microsoft: An Economic Analysis." *Did Microsoft Harm Consumers? Two Opposing Views*. Washington, D.C.: AEI-Brookings Joint Center for Regulatory Studies, 2000. 1-44. Print.

"Google Inc. Stock Price - NASDAQ.com." *NASDAQ Stock Market*. Web. 02 Apr. 2011. <http://quotes.nasdaq.com/asp/SummaryQuote.asp?symbol=GOOG>.

Gruber, John. "Apple, Adobe, and Flash." *Daring Fireball*. 25 Jan. 2010. Web. <http://daringfireball.net/2010/01/apple_adobe_flash>.

Hahn, Robert, and Robert Litan. Foreword. *Did Microsoft Harm Consumers? Two Opposing Views*. By David S. Evans. Washington, D.C.: AEI-Brookings Joint Center for Regulatory Studies, 2000. Print.

Hamilton, Naomi. "The A-Z of Programming Languages: JavaScript - A-z of Programming Languages." *Computerworld*. 31 July 2008. Web. 19 Dec. 2010. <http://www.computerworld.com.au/article/255293/a-z_programming_languages_javascript/?pp=2&fp=4194304&fpid=1>.

Hamilton, Naomi. "The A-Z of Programming Languages: JavaScript - A-z of Programming Languages." *Computerworld*. 31 July 2008. Web. 19 Dec. 2010. <http://www.computerworld.com.au/article/255293/a-z_programming_languages_javascript/?pp=2&fp=4194304&fpid=1>.

Hardmeier, Sandi. "The History of Internet Explorer." *Microsoft Corporation*. 25 Aug. 2005. Web. 14 Sept. 2010. <http://www.microsoft.com/windows/IE/community/columns/historyofie.mspx>.

Herman, Jeff. "A Process for Creating the Business Case for User Experience Projects." Proc. of CHI EA '04, Vienna. Web.

Hewitt, Joe. "Status Updates, 28-30 April 2010." *Twitter*. Apr. 2011. Web. <http://twitter.com/joehewitt/>.

"A History of Windows." *Microsoft Corporation*. Web. <http://windows.microsoft.com/en-us/windows/history>.

Introna, Lucas D., and Helen Nissenbaum. "Shaping the Web: Why the Politics of Search Engines Matters." *The Information Society* 16.3 (2000): 169-85. Print.

Jobs, Steve. "Thoughts on Flash." Web log post. *Apple Inc.* 29 Apr. 2010. Web. <http://www.apple.com/hotnews/thoughts-on-flash/>.

Katz, Yehuda. "The Web Doesn't Suck. Browsers Are Innovating." Web log post. *Yehuda Katz*. 30 Apr. 2010. Web. <http://yehudakatz.com/2010/04/30/the-web- doesnt-suck-browsers-are-innovating/.>.

Keith, Jeremy. *HTML5 for Web Designers*. New York, NY: Book Apart, 2010. Print.

Kotamraju, Nalini P. "Keeping Up: Web Design Skill and The Reinvented Worker." *Information, Communication & Society* 5.1 (2002): 1-26. Print.

Krill, Paul. "JavaScript Creator Ponders Past, Future | Developer World - InfoWorld." *Developer World*. InfoWorld, 23 June 2008. Web. 19 Dec. 2010. <http://www.infoworld.com/d/developer-world/javascript-creator-ponders-past-future-704?page=0,0>.

Kristol, D., and Lou Montulli. "HTTP State Management Mechanism." *Internet Engineering Task Force*. Feb. 1997. Web. <http://www.ietf.org/rfc/rfc2109.txt>.

Kristol, David M. "HTTP Cookies." *ACM Transactions on Internet Technology* 2.1 (2001): 151-98. Print.

Lewis, Michael. *The New New Thing: a Silicon Valley Story*. New York: W.W. Norton, 2000. Print.

Libicki, Martin C., James Schneider, David R. Frelinger, and Anna Slomovic. *Scaffolding the New Web: Standards and Standards Policy for the Digital Economy.* Santa Monica, CA: Rand, 2000. Print.

Lie, Hakon W. "Cascading Style Sheets." Thesis. University of Oslo, 2005. Web. 28 Apr. 2011.

Lohr, Steve. "Part Artist, Part Hacker And Full-Time Programmer." *New York Times* 9 Sept. 1996. Web. 4 Dec. 2010. <http://www.nytimes.com/1996/09/09/business/part-artist-part-hacker-and-full-time-programmer.html?pagewanted=all>.

Manjoo, Farhad. "7 Ways Larry Page Is Defining Google's Future." *Fast Company*. 16 Mar. 2011. Web. <http://www.fastcompany.com/magazine/154/the-quest.html?page=0%2C1>.

Mayo, Keenan, and Peter Newcomb. "How the Web Was Won." *Vanity Fair* July 2008. *Vanity Fair*. Web. <http://www.vanityfair.com/culture/features/2008/07/internet200807>.

Microsoft Corporation. "Microsoft's Tradition of Innovation." 25 Oct. 2002. Web. <http://www.microsoft.com/about/companyinformation/ourbusinesses/profile.mspx>.

Montulli, Lou, and Dave Kristol. "HTTP State Management Mechanism." *Internet Engineering Task Force*. Feb. 1997. Web.

Mozilla Corporation. "Mozilla 2006 Financial FAQ." *Mozilla Project*. Web. <http://www.mozilla.org/foundation/documents/mozilla-2006-financial-faq.html>.

Muller-Brockmann, Josef. *Grid Systems in Graphic Design: a Visual Communication Manual for Graphic Designers, Typographers and Three Dimensional Designers = Raster Systeme Fur Die Visuelle Gestaltung : Ein Handbuch Fur Grafiker, Typografen Und Ausstellungsgestalter*. Sulgen: Verlag Niggli, 1996. Print.

O'Reilly, Tim. "What Is Web 2.0 - O'Reilly Meia." *O'Reilly Media*. 30 Sept. 2005. Web. <http://oreilly.com/web2/archive/what-is-web-20.html>.

Oudshoorn, Nelly, and T. J. Pinch. *How Users Matter: the Co-construction of Users and Technologies*. Cambridge, MA: MIT, 2003. Print.

Pattinson, Hugh, and Linden Brown. "Chameleons in Marketspace: Industry Transformation in the New Electronic Marketing Environment." *Internet Research* 6.2/3 (1996): 31-40. Web.

Peng, Weihong, and Jennifer Cisna. "HTTP Cookies – a Promising Technology." *Online Information Review* 24.2 (2000): 150-53. Print.

Pilgrim, Mark. "Why Do We Have an IMG Element? [dive into Mark]." Web log post. *Dive into Mark*. 28 May 2010. Web. 14 Sept. 2010. <http://diveintomark.org/archives/2009/11/02/why-do-we-have-an-img-element>.

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.

Quinn, Christine A. "The Name Assigned to the Document by the Author. This Field May Also Contain Sub-titles, Series Names, and Report Numbers. From Grass Roots to Corporate Image--The Maturation of the Web." *CAUSE/EFFECT* 18.3 (1995): 49-51. Web.

Raggett, Dave. "HTML Tables." *Internet Engineering Task Force*. 1996. Web.

Reid, Robert. *Architects of the Web: 1,000 Days That Built the Future of Business*. New York: John Wiley & Sons, 1997. Print.

Rusli, Evelyn M., and Verne G. Kopytoff. "Investing Like It's 1999 - NYTimes.com." Web log post. *DealBook - NYTimes.com*. 27 Mar. 2011. Web. 28 Apr. 2011. <http://dealbook.nytimes.com/2011/03/27/is-it-a-new-tech-bubble-lets-see-if-it-pops/>.

Russell, Alex. "View-Source Is Good? Discuss." Web log post. *Infrequently Noted*. 7 Jan. 2010. Web. 14 Sept. 2010. <http://infrequently.org/2010/01/view-source-is-good-discuss/>.

Russell, Alex. "View-Source Is Good? Discuss." Web log post. *Infrequently Noted*. 7 Jan. 2010. Web. <http://infrequently.org/2010/01/view-source-is-good-discuss/>.

Sage, Alexandria. "Amazon's Projected Cost of Growth Hits Shares Hard." *Reuters*. 27 Jan. 2011. Web. <http://www.reuters.com/article/2011/01/27/us-amazon-idUSTRE70Q5JU20110127>.

Seibel, Peter. *Coders at Work: Reflections on the Craft of Programming*. [New York]: Apress, 2009. Print.

Siegel, David. "The Web Is Ruined and I Ruined It." *World Wide Web Journal - Special Issue on XML: Principles, Tools, and Techniques* 2.4 (1997). Web.

Sink, Eric. "Memoirs From the Browser Wars." *Eric Sink's Weblog*. 15 Apr. 2003. Web. 14 Sept. 2010. <http://www.ericsink.com/Browser_Wars.html>.

Sivonen, Henri. "Activating Browser Modes with Doctype." Web log post. *Henri Sivonen's Page*. Web. <http://hsivonen.iki.fi/doctype/>.

Spinello, R. A. "Competing Fairly in the New Economy: Lessons from the Browser Wars." *Journal of Business Ethics* 57.4 (2005): 343-61. Print.

"Steve Jobs: The Next Insanely Great Thing." Interview by Gary Wolf. *Wired* 02 Apr. 1996. Print.

Taivalsaari, Antero, Tommi Mikkonen, Dan Ingalls, and Krzysztof Palacz. *Web Browser as an Application Platform: The Lively Kernel Experience*. Rep. no. SMLI-TR-2008-175. Menlo Park: Sun Microsystems Laboratories, 2008. Print.

Treisman, Anne M., and Gary Gelade. "A Feature-integration Theory of Attention." *Cognitive Psychology* 12.1 (1980): 97-136. Print.

"Visualizing Usage of the Firefox Menu Bar « Alex Faaborg." *The Mozilla Blog*. 23 Mar. 2010. Web. 14 Sept. 2010. <http://blog.mozilla.com/faaborg/2010/03/23/visualizing-usage-of-the-firefox-menu-bar/>.

Web History Project. "Index of /www.lists." Web. <http://1997.webhistory.org/www.lists/>.

Wright, Alex. "New Search Technologies Mine the Web More Deeply." *The New York Times*. 22 Feb. 2009. Web. <http://www.nytimes.com/2009/02/23/technology/internet/23search.html>.

Zeldman, Jeffrey. *Designing with Web Standards*. Indianapolis, IN: New Riders, 2003. Print.

Zhao, Weiquan, David Kearney, and Gianpaolo Gioiosa. "Architectures for Web Based Applications." Proc. of 4th Australasian Workshop on Software and Systems Architectures, Monash University, Melbourne. 2002. Web.

Zittrain, Jonathan. "The Un-Microsoft Un-Remedy: Law Can Prevent the Problem That It Can't Patch Later." Reading. Connecticut Law Review Spring Symposium. 26 Mar. 1999. Web. 4 Apr. 2010. <http://cyber.law.harvard.edu/zittrainmsdoj.pdf>.