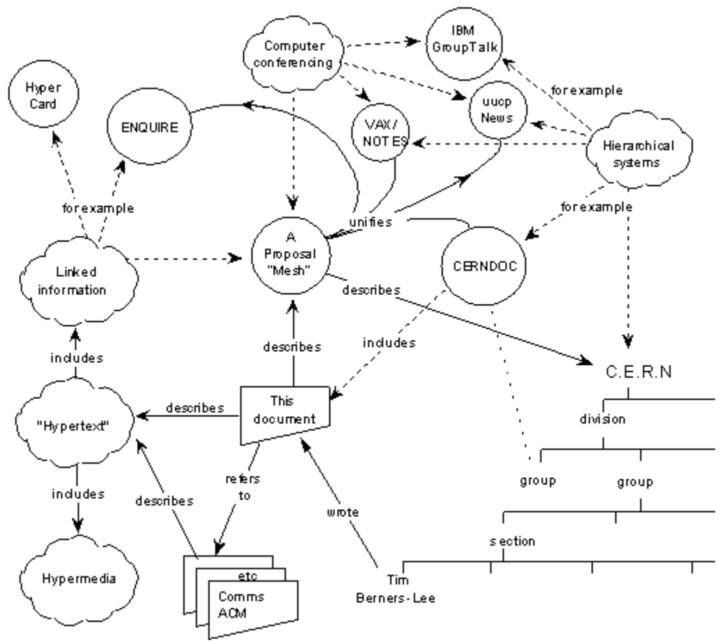
The World Wide Web as an Engineering Paradigm



Matthew Lee
Andrew Montgomery
Steven Shapiro
Veeral Shah
Qian Wang

December 15, 2000

6.933/STS.420 – The Structure of Engineering Revolutions
Professor David Mindell and Professor George Pratt

Introduction

"You see things and say 'Why?'
but I dream things that never were, and I say 'Why not?'"

George Bernard Shaw (1856 – 1950)

The World Wide Web is a compelling example of an engineering paradigm. The adoption of the Hypertext Markup Language (HTML) and the Hypertext Transfer Protocol (HTTP) as standards represent the acceptance of this new paradigm—a revolution in information sharing. This work details the technical evolution of HTML and HTTP, from their inception to their acceptance as the technological foundation of the World Wide Web.

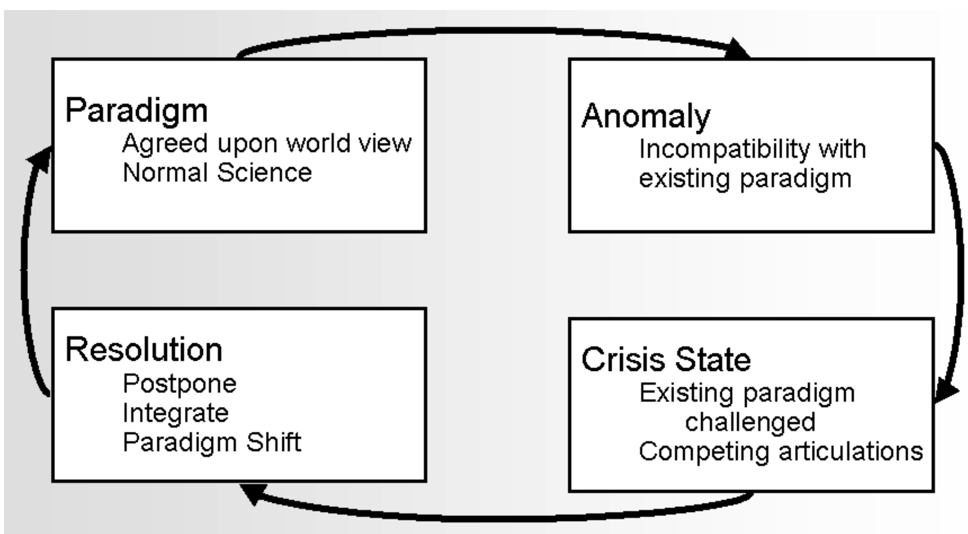
This project history first equates the Web's development an engineering paradigm, with a discussion of how it departs from the Kuhnian model of scientific paradigms. Next, a history of the important events preceding the development of the Web sets the stage for an understanding of the background in which its creation took place. This will help to illuminate the presence of particular anomalies. We will then examine the period of crisis that would ensue—the question of how to share information across multiple platforms. In the presence of this crisis, competing articulations and solutions arose—such as Gopher, WAIS, and Guide. Finally, this history delineates the rapid acceptance of the World Wide Web—the adoption of a new paradigm.

By no means did the World Wide Web signal the advent of online communications. The core technologies that the Web is based upon, such as TCP/IP and hypertext, existed for years and in some cases decades before the Web was invented. Files were transferred with FTP, network news was read through NNTP, and network packets circled the globe through the Internet. Similar technologies to the Web existed as well, including Gopher, a system developed at the University of Minnesota.

Despite all of these perfectly functional technologies, the Internet was fraught with incompatibilities, as data was stored in a myriad of often-incompatible formats and the usage of protocols was inconsistent across different platforms. For example, the simple act of retrieving a file could prove troublesome. First, one would have to locate a remote server that the file would be hosted on, through a primitive search service such as Archie or by being told about a server from a colleague. Then the file would have to be retrieved via FTP, which involves a series of directory navigation commands and some protocol negotiation as well. Finally, the file would have to be viewed locally, and the viewer might be incompatible or nonexistent for a particular format, especially when the file was created on another platform. Tim Berners-Lee invented the Web in part to resolve these incompatibilities, relying on a small set of simple protocols to share data in a platform-independent manner. This is the story of the Web.

Scientific Revolutions

Thomas Kuhn, in his seminal work "The Structure of Scientific Revolutions," outlined the development of a paradigm:



1) **Paradigm / Pre - Paradigmatic state**: At this time, there may or may not be a conceptual framework by which people assessed the day's ideas. This framework represents the "agreed upon world view" at the time. Within this state, the progress that takes place is "normal science:" these developments are incremental, and do not challenge the prevailing paradigms.
2) **Anomaly**: An incompatibility with the existing paradigm arises. Evidence or observations appear to contradict the currently understood framework.

3) **Crisis State**: This stage is the reaction to the anomaly. The period can be characterized by confusion, and the existing paradigm is challenged. Competing articulations emerge, to try to explain the newly found evidence.
4) **Resolution**: The resolution of the crisis may take one of three forms:
    a. *Postponement* – The anomaly is put aside because it is not yet well understood.
    b. *Integration* – The new phenomenon is sometimes found to be compatible with the previously existing paradigm, and the two are integrated into a fuller, more complete paradigm.
    c. *Paradigm Shift* – This is the most momentous of the three types of reactions. Out of the competing solutions emerges an undisputed new consensus. It is this consensus that defines the new conceptual framework as the prevailing paradigm.

Engineering Revolutions

The characteristics of engineering revolutions differ along some important metrics from the Kuhnian model, which is more suitable for describe scientific change. Noted here are several key differences.

Technology Driven: The primary difference between scientific and technological revolutions is, quite simply, the artifacts at hand. In scientific revolutions, what previously exists is a framework upon which to base thought processes and research. In contrast, technological revolutions involve technological artifacts, with a newer 'artifact' dominating. In these cases, both artifacts work properly, and provide some benefit to their users, but the mere existence (and growing usage) of the newer technology instead drives the revolution.

Not Fundamental or Global: Technological revolutions are characterized by a 'better way of doing things.' Unlike the acceptance of a broad scientific paradigm, the acceptance of a new engineering paradigm does not have fundamental implications on the whole world.

Consensus is Not Necessarily Universal: Often times a new engineering paradigm is not universally accepted. It is only important that the new framework dominates its field.

The rest of this paper explores the advent of the World Wide Web, providing a chronicle of its development within the defined technological revolution framework.

The Existing Information Sharing Paradigm

We have just introduced our concept of the structure of engineering revolutions. Any such revolution must begin with either an existing paradigm or a pre-paradigmatic state that resembles the crisis state we have discussed. The engineering revolution that is the focus of this paper is that of the World Wide Web and its revolution of the existing information-sharing paradigm. Before discussing the Web itself, however, we must first examine the paradigm immediately preceding the Web, the new technologies which arose that presented an anomaly in the paradigm, the crisis state that was created by this anomaly and the competing solutions that arose to solve the crisis. We must understand what occurred before the Web in order to understand why the Web was developed and why it is *the* way to share information electronically today.

Computer systems prior to the 1960s consisted mostly of mainframes at large research and educational institutions and businesses. No one had conceived of a personal computer as we understand them today. The systems first used punch-cards as a method of input. Each system had a designated operator that would feed it the cards dictating operations that various users wanted performed. Only one set of cards could be inputed at a time and the system could only do one set of operations. But, the systems were very large and relatively fast and could handle much more work. Thus, timesharing of the systems began, in which a terminal with a wire connecting it to the mainframe could be placed on a users desk and punch cards were eliminated. Each system used its own operating system and protocols. There was no need for the systems to be compatible with each other, as there was no network to connect them. Data was thus confined to the local machines and information could only be shared within the institution which housed the mainframe. Vannevar Bush, director of the US Office of Scientific Research and Development under President Roosevelt, wrote an article in 1945 titled *As We May Think* published in *Atlantic Monthly*. These were the some of the first thoughts that were published thinking about the power of computers in information sharing. His “Memex” sounds a great deal like today’s personal computers and the user’s interaction with the system very much resembles that of hypertext, even though no one would “invent” either technology for decades.

This leads us to the three technologies that created an anomaly in the information sharing paradigm. Collaboration and hypertext systems would give people a way to organize and share their information. Pervasive networking would allow people to connect their systems together over large distances. Personal computers would allow any person, at the office or at home, to share their information. Not until all three of these technologies were developed was there really an anomaly in the information sharing paradigm. We will now examine each of these technologies in more detail in order to understand this anomaly.

New Technologies Cause an Anomaly

Collaboration and Hypertext Systems

If it was not for Vannevar Bush’s provocative article, *As We May Think*, Doug Engelbart may not have invented the mouse or formed the idea of the first hypertext system. Engelbart first stumbled across the Vannevar Bush article and recalls that it had a profound effect on him. "I remember being thrilled," he says. "Just the whole concept of helping people work and think that way just excited me. I never have forgotten that."[1] He was inspired and began to work on his oN-Line System (NLS) in 1963 and first publicly demonstrated it in 1968 at the Fall Joint Computer Conference in San Francisco. The system really was revolutionary for its time; it incorporated a graphical display, video

[1] How the Web was Born, p. 94

conferencing, hyperlinks for organizing information, and the computer mouse (which he invented for the project) for navigation. See Figure 1, a screen shot of Engelbart's demonstration of NLS in 1968. NLS was a completely successful implementation of what Engelbart had imagined and people congratulated him for it. However, the same people just could not see the purpose of it given their view of the role of computers: large mainframes using punch cards. Engelbart was a man far ahead of his time. Decades later, when asked about the lack of acceptance for NLS, he says "You can't bring stuff out against the prevailing paradigms."[2] It would be more than twenty years before many of Engelbart's ideas would be used to force a paradigm shift.

Figure 1 Engelbart demonstrating NLS in 1968.

Other collaboration and hypertext systems were also in development during the 1960s. In fact, Ted Nelson would coin the term "hypertext" during a presentation at the 1965 Association for Computing Machinery conference in Pittsburgh. Ted Nelson would be one of the first insightful individuals to develop Doug Engelbart and Vannevar Bush's shared vision. Nelson first developed a system called Xanadu, which he described as a computerized version of Bush's Memex. Xanadu will be discussed in more detail shortly. Nelson went to work with Andy Van Damm and some of his students at Brown in 1967 to create the second hypertext system, the Hypertext Editing System (HES). This was not an entirely significant hypertext system, but more importantly Andy Van Damm and his students continued, after Nelson stopped collaborating in 1968, to work on a new system called the File Retrieval and Editing System (FRESS). Although FRESS had no networking (the first ARPANET node was not even online yet), it had two things that later hypertext systems would find valuable. First, it had bi-directional links that allowed for the user to control the link structure without modifying the original document content.

[2] Doug Engelbart interview

The second was a form of metadata that allowed keyword searching of links. Other systems that were to follow, such as Microcosm and Intermedia, borrowed from these ideas separating the links from the content. FRESS proved to be a relative success in the classroom for which it was designed. However, FRESS and other similar systems would not solve the information sharing crisis. The nature of these systems was to separate the links from the original source documents and maintain them in a central database. This database would have to be continuously maintained in order to retain the consistency of the bi-directional links. This, in turn, would prevent FRESS and other similar systems from scaling to a large (especially global) scale.

Ted Nelson's Xanadu project continued for over two decades after its original inception. It provided a framework for information sharing, culture and management. Xanadu's biggest difference from other systems of the time was its micropayment scheme. Xanadu's creator was not a member of the computer science community [say what he was here]. This may have been what caused Nelson to heavily focus on giving credit to the author of every bit of information. It is suspected that Xanadu never really became a complete system because of its visionary nature. A product must be concrete. Xanadu presented itself almost as an entirely different way of life. Nelson's writing on Xanadu, "Literary Machines," made references not only to information management, but what people would think and how they would dress. Xanadu was just too far reaching to be seriously considered for a viable solution to the engineering crisis at hand.

Pervasive Networking

On October 4th 1957, the Soviet Union made history when they launched Sputnik I into orbit. President Eisenhower declared that the United States would never again be caught off guard by the Soviet Union and went to Congress on January 7th, 1958 to request funding for a new research and development organization. Congress quickly approved funding for the new Advanced Research Projects Agency (ARPA) and, by 1962, ARPA's long-term goals had been refined. The two major parts of ARPA involved ballistic missile defense and nuclear test detection. However, there was a small office, using less than 10 per cent of ARPA's budget, called the Information Processing Techniques Office (IPTO) that by 1965 was immersed in research for advanced computing. Computer time-sharing among many different systems across the country was the solution to the lack of funding the IPTO had and the multitude of requests from computer science departments who each wanted to build very powerful computers. By sharing the computer resources, more could be done. Thus, the IPTO began work on a nationwide network to connect the various systems. By late 1966, plans for the ARPA Computer Network, or ARPANET, had begun.

It was not immediately clear how the network should be built. Initial ideas from the project's director, Larry Roberts, focused on just having each computer call up all the other computers using the existing phone lines to exchange information. This would force each computer to understand the operating system of all the others. There are obviously many problems with this approach, as the network could not scale very well or be flexible. Wesley Clark, an ex-colleague of Roberts, came up with the idea of having small computers on the network to route data around, or sub-nets. This would mean the large computers on the network would only have to know one protocol or language to talk to each other – that of the small routing computers. The small computers were named Interface Message Processors (IMP) and their use was essential to the success of ARPANET. Another important part of the design or ARPANET was that it used packet-switching. This new idea, developed independently by both Paul Baran and Donald Davies, cuts messages up into small chunks of the same size for transmission across the network. It allowed a network to exist that would be able to survive machine failure on one part of the network; the packets could just be rerouted through different parts of the network.

UCLA was the chosen location for the first IMP and the second was put into place at the Stanford Research Institute (SRI) in October 1969 and the first packet-switching computer network was established. ARPANET grew quickly, adding about one host per month. In 1972, the first email was sent across

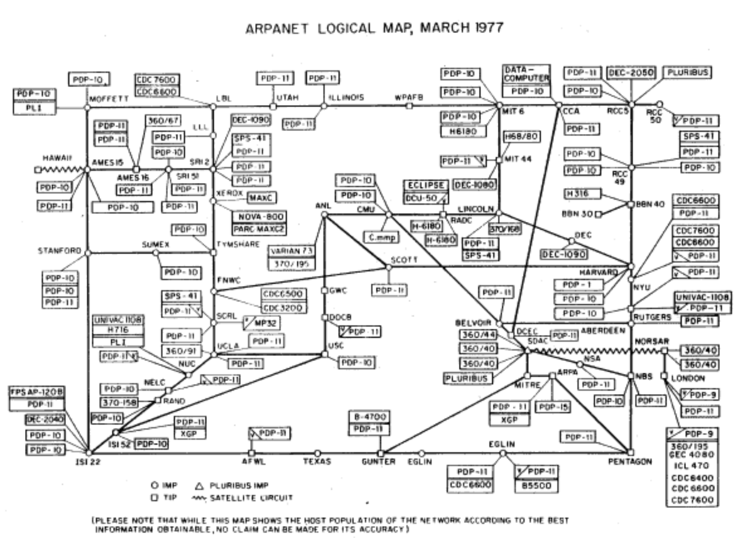


Figure 2 The ARPANET in March 1977.

ARPANET and the network was publicly demonstrated for the first time at the first International Conference on Computer Communication (ICCC) in Washington, DC. See Figure 2 for a picture of ARPANET in 1977.

The first protocols on the ARPANET were hastily thrown together; they were telnet and FTP. Telnet allowed someone sitting at a terminal to login to a computer at another site. FTP, File Transfer Protocol, allows files to be transferred from one computer to another. These two protocols were a good start, but hardly sufficient for the network. In June 1973, Vint Cerf, Bob Kahn, Bob Metcalfe and Gérard Le Lann started to work on the design of a host-to-host protocol for inter-networking. They called it the Transfer Control Protocol (TCP). TCP proved to be a robust host-to-host protocol that allows reliable communication over unreliable networks. In 1978, TCP was combined with another protocol, Internet Protocol (IP), to form TCP/IP. IP controlled the routing of packets around the network. Thus, with the introduction of the Domain Name System (DNS) by Paul Mockapetris and the adoption of TCP/IP by ARPANET in 1983, the Internet as it is known today was born.

This pervasive network, the Internet, allowed any network of computers to be connected to any other network of computers. People could now share information electronically, using a universally agreed-upon protocol. But, in the early 1980s, computers were still primarily at large institutions and there was a relatively small group of users. Not until the rise of personal computers would there by a true crisis in the paradigm of information sharing.

Personal Computers

Computers used in the existing information sharing paradigm were massive in both size and price. Only large institutions could afford and support them. But, the invention of the transistor in 1947, the integrated circuit in 1958, the microprocessor in 1969, and magnetic memory in 1972, set the stage for computers to become smaller and cheaper. They would become a part of most people's daily lives– a PC would soon sit on

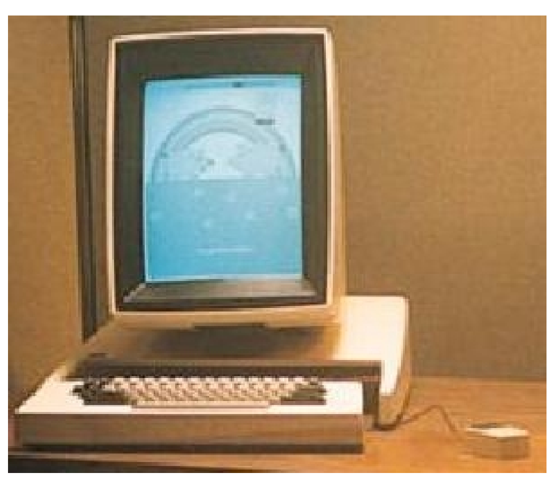


Figure 3 The Xerox PARC Alto

their desk at work *and* at home.

The Xerox Palo Alto Research Center (PARC) began work on its Alto computer in 1973. It was the first computer designed specifically for use by non-computer scientists. It actually was the first system to use many of Doug Engelbart's ideas, most notably the mouse. It had a graphical user interface (GUI) with windows, menus and icons. It looked very much like today's PCs. The machine that evolved from the Alto, the Xerox Star, was the first system that Xerox tried to sell, but very few were actually sold because of its $18,000 price tag.

In January 1975, an almost completely different system was released. The Altair 8800, released by Micro Instrumentation Telemetry Systems (MITS), sold for $397. Users had to assemble the system themselves and the system was essentially a series of switches and lights. It was completely unclear how it even worked. But, many were sold to those who were curious about computers and had never had the chance to be exposed to them. It inspired them to be creative and innovate. Bill Gates and Paul Allen, for example, saw the Altair and formed Microsoft to sell software that allowed BASIC programming on the system. Steve Wozniak saw the Altair and was determined to build something better; the cost of microprocessors and memory had fallen dramatically. Wozniak joined forces with Steve Jobs to form Apple Computer and they released the Apple II at the West Coast Computer Faire in May 1977, selling it for a more reasonable $1300. Many Apple II's sold quickly and many companies started to develop software for it.

Figure 4 The IBM PC-XT

The early 1980s saw the real birth of the personal computer as it is thought of today. In 1981, IBM launched its PC, selling for $2,800. Businesses loved the PC. IBM's decision to allow clones ensured rapid growth of the technology. Microsoft wrote the operating system, MS-DOS, and would later write the Windows operating system. During this same timeframe, Apple was releasing its next computer. In 1984, Apple released its first Macintosh computer, selling it for $2,495. It was the first computer that brought the technology that Xerox had developed based on Engelbart's ideas to the masses. The GUI stunned people and the Macintosh quickly developed a strong following amongst desktop publishers, something that it still maintains today. The IBM PC and the Macintosh changed the face of personal computing forever.

A Crisis Develops

The exponential growth of personal computers into the 1990s signaled the establishment of the third technology that was needed for an anomaly in the information sharing paradigm. Recall that an anomaly occurs in an engineering paradigm when an incompatibility with the existing paradigm arises. In this case, information that was previously used by few individuals and stored on isolated systems suddenly had the potential to be used by many people and on any system. The three technologies – collaboration/hypertext systems, pervasive networking, and personal computers created this anomaly. But, there was no single solution to solve the problem created by this anomaly. No system was commonly accepted that allowed people to collaborate across a network using their PC. Until such a solution was developed, the anomaly brought about a crisis in the information sharing paradigm. Competing articulations for solving the anomaly is characteristic of a crisis state and we will now discuss those competing solutions.

Any solution to this crisis state would have to have three characteristics for it to be successful. First, it must possess the ability to discover the information that is desired. The Internet allowed all the networks in the world to be connected, but provided no real means for finding desired information scattered across the millions of machines that were connected. Secondly, any solution would have to be able to retrieve the information regardless of the operating system and software being used. A standard protocol implemented on all platforms would be essential for success. Finally, the new system would have to be able to display the information uniformly once it was received, again regardless of the platform being used. Several solutions were developed that shared all of three of these characteristics.

Competing Solutions

Some would argue that creativity is at its peak during an engineering crisis state. In the case of the crisis being described in this paper, we will see that this is the case. There were many competing articulations that attempted to solve the discovery, display, and retrieval crisis in information sharing described above. However, these solutions solved the problem in a variety of creative and differing manners. There were more competing articulations than can adequately researched for the scope of this paper, but several, such as WAIS, Archie, Guide, and Gopher, will be discussed in the following sections. Two solutions are particularly relevant to development of the World Wide Web project as well as the question of 'what could have been if only...', Office Workstation Limited's Guide and Mark McCahill's Gopher.

Wide Area Information Servers

Wide Area Information Servers (WAIS) could be construed as the opposite of Xanadu. WAIS was concrete, concise and focused with a commercial marketplace. It was the first successful information management system that was a commercial venture from the start. Brewster Kahle, then working for the supercomputing company Thinking Machines, was the head of the project. WAIS was similar to a modern day search engine. It's key benefit was its ease of use and relatively intuitive interface. Figure 5 below is one example screenshot of the first WAIS client from a paper written in part by Brewster Kahle published in *Electronic Networking* the journal.[3] Users could submit their query in plain English. Since files were described by keywords, media such as video or sound files could be searched as well as text. WAIS proved very successful and soon spun off into WAIS Inc. which was sold in 1995 to America Online for $15 million. WAIS Inc. was one of the first Internet companies and proved to many skeptics that an Internet company was possible. WAIS did not become the new paradigm because it was missing one key ingredient, a humanitarian vision. WAIS Inc. and Brewster Kahle's vision were

[3] B. Kahle et al., Wide Area Information Servers : An Executive Information System for Unstructured Files, in: *Electronic Networking: Research, Applications and Policy*, Vol. 2, No. 1 (Meckler, New York, 1992) 59-68.

not compatible with a free world-wide solution. The company was merely creating a product for profit.

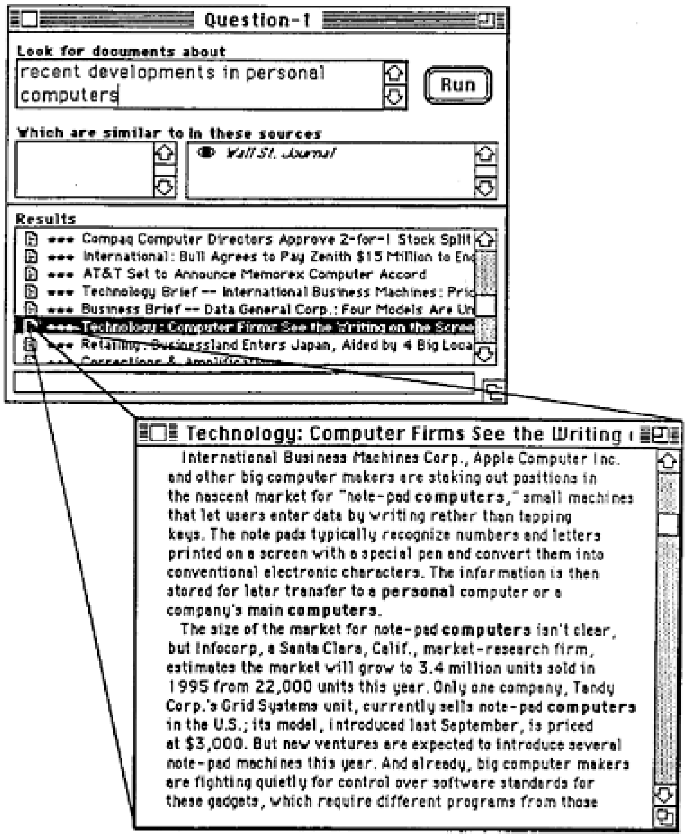


Figure 5 A screenshot of the first WAIS client demonstrating the ease of use for the time.

Archie, Prospero and FTP

Telnet and FTP were the first protocols on ARPANET and Neither protocol's clients were robust or easy to use. This is where Archie came into play as a possible solution to the ongoing crisis. In order for users to find programs on the Internet in 1983, they would be forced to conduct manual searches of known FTP servers. Even then, the filenames of these programs may not yield useful descriptions. Alan Emtage, a student at McGill University in Montreal, Canada, sought to alleviate some of the mundane searching with Archie in 1983. Archie is a few shell scripts that automate this search. As Archie quickly became popular, Clifford Neuman, a Ph.D. student at the University of Washington, saw his chance to improve upon Archie by integrating it with his distributed filesystem, Prospero. Prospero was used to organize Archie's files as well as expedite the discovery process. This system quickly became very popular because of the growing urgency for a new paradigm. The Prospero-Archie-FTP system adequately solved discovery, display and retrieval problems, but it did not become the new paradigm. "Only hardcore users knew about FTP."[4] This system lacked the necessary easy of use for the general computer illiterate population. In addition, although it was a breakthrough for its time, the system left much in the way of improvement especially with respect to its display of information. Long filename lists with descriptions are hardly comparable to the media rich interface of the Web as we know it today.

Real Solutions

Two contemporaries of the web deserve a more detailed look in this paper because of how close they actually became to becoming the new paradigm. Interviews with the systems parents and corroborative documents shed light onto this rarely documented turn of the Web. Tim Berners-Lee, the undisputed father of the World Wide Web, approached both Office Workstations Limited (OWL) and the Gopher Team to try to convince them to adapt their systems to form to his humanitarian vision of a single global information space.

Guide

Guide was first developed as an academic project by Peter Brown in 1983 at the University of Kent in the United Kingdom. This academic version of the hypertext system Guide worked a little differently than its later commercialized counterpart:

> The way it works is that when you hit a button it replaces that button in-line so there is a sort of metaphor in which you have an expanding document; you hit a button and that button expands, in-line, so it remains in the context where it was in the whole document. If you undo it, it goes back to the previous [context].[5]

Ian Ritchie, the founder of Office Workstations Limited (OWL), purchased Guide from the Peter Brown and the university with a sole license to the system.

OWL released their version of Guide for the Macintosh in August 1986, winning

[4] Mark McCahill phone interview.
[5] Peter Brown phone inteview.

the honor of becoming the first commercially successful hypertext system. Almost exactly a year later, OWL released an identical version of Guide for the PC. However, Apple Computer, Inc. forced OWL to concentrate on their newly released PC version of Guide. In the middle of 1987, Apple released their hypertext system, Hypercard, free with every Macintosh. Ian Ritchie notes in reference to OWL's shift to the PC market, "You can compete with other things, but you can't really compete with free."[6] This idea holds true in many other cases described in this paper.

OWL's implementation of Guide was extremely robust and much more advanced than Berners-Lee's World Wide Web. Guide used a markup language called the Hypertext Markup Language (HML) not to be confused with Berners-Lee's HTML. Ian Ritchie describes HML:

> That was a version of SGML with added tags to indicate things like a button or a button source or a point, headers and body text, so forth.. And by marking up your documentation in that way you could automatically generate a hypertext display. And you could go from document to document. And if your documents were on a file server you could go across a network to them.. We did that… I mean the interesting thing now is that everybody is really adopting XML. But XML is.. essentially contains the kind of structures that we and other people had done in the late 80's with our markup languages. I mean our markup language, old HML, was more than just a hypertext markup language. It actually was a content structure language as well. It described for you, what was a heading.. what was a subheading.. what was an author's name.. what was an abstract.. what was a caption.. Yeah, that's the kind of thing that XML does. In fact there's evidence of that in a variant we were doing for <sounded like Verfe Ville Industry>. Cause we put the first thing that Ford Motor adopted for all of their technical publishing in North America. And in fact, later, they also were adopted by General Motors. So we had both General Motors and Ford. And there we just adopted their automobile markup language. They had a language they'd determined for marking up documents for automobile information.
>
> And that's.. All that stuff was very similar to what people today call XML.[7]

OWL even produced Guidex (originally IDEX), which was a solution for a

[6] Ian Ritchie phone interview.
[7] Ian Ritchie phone interview.

particular client which was a networked version of Guide. Ian Ritchie even described a similar version of Guide for a client that worked over the Internet! It is mind boggling to look at this little-known hypertext system that was much more advanced than the famous World Wide Web before the Web was really started.

Tim Berners-Lee was looking for a jump start on his Web project in an already existent system. He finally saw one that peaked his interest when Guide was demonstrated at the first European Hypertext Conference at Versaille in Paris in 1990. Ian Ritchie remembers it clearly:

> What Tim wanted was a graphical user interface to make that more user friendly. And we had a chat about it.. Actually, I mean this was hardly a summit meeting.. He and I met in the bar at the hypertext conference in Versaille and he mentioned this project that he was working on called the World Wide Web. I remember thinking it was a rather grand name.
>
> Yeah, it did seem like a bit of a grand name at the time. But he said that they were working on that stuff and they were taking off great well.. and he wondered if we might want to, you know, share in some of this work. He was particularly interested in the work we'd been doing. He'd read about the work we'd been doing in GuideX because he heard we had a markup language. So I agree that I would send him details of that. And we sent him various specs and so forth of the markup language. We had a sort of exchange of correspondence.[8]

However, Guide and Guidex were not public domain, and OWL was a for-profit company. CERN was in no position to purchase external engineering as they were devoting most of their budget to particle accelerator projects and other expensive physics equipment. If CERN had purchased Guide from OWL it quite possibly could have become what we think of as the World Wide Web today. However, Guide was much more advanced than the Web, so some of the well established problems of the Web today (such as dead links) may have already been solved.

Gopher

In the late 1980's all major universities had embraced the Internet and begun to

[8] Ian Ritchie phone interview.

digitize various content. These institutions were plagued by the same discovery, display and retrieval crisis as the rest of the networked community. In the early 1990's, Campus Wide Information Systems (CWIS) seemed to be the obvious solution. These systems were very different, but they all had a common goal: To organize the vast microcosm of the university's information space.

The University of Minnesota was no exception. A lot of people had an interest in participating in the university's effort, so a committee was formed to design their ideal system. Mark McCahill, an employee at the university's computing center, was eventually given the task of managing the project. In short, McCahill and his lead developer, Farhad Anklesaria, felt the messy decree delivered by the committee was unacceptable and followed their own design. Their CWIS was named Gopher, after the university's mascot, and McCahill and Anklesaria went to the committee in 1991. Gopher met with enormous resistance from the University committee because it did not follow their decree. The University of Minnesota rejected Gopher. However, the system was still released to the public domain in June of 1991. Gopher quickly became the Web's biggest contemporary competitor and was the first solution to the impending crisis that really opened the internet to anyone.. It was simple, intuitive and well-structured. A hierarchy of menus whose items can be submenus or an object (like a document, picture or any other object). "The information for any given item is enough to identify the server it's on" (p.138 h.w.b.) This feature was integral to the success of Gopher because it meant the information did not have to be stored on the computer you were using. One thing had to be added before Gopher's preliminary release, full-text database searching.

Tim Berners-Lee also recognized the value of Gopher as a world-wide system and approached the Gopher team at the 1991 Internet and Engineering Task Force (IETF) meeting..

> Tim was all excited and said 'Look at this stuff I came up with! We should work together because we're doing a lot of the same things.' But when I got back and looked at it some more, I said, 'Shit, I don't work this way!'[9]

Mark McCahill was quick to reject Tim Berners-Lee's ideas for one very good

[9] Mark McCahill phone interview.

reason: the Web project lacked structure. In fact, from McCahill's description, the Gopher team would have rejected any idea to migrate to a hypertext system where objects can be randomly associated. Gopher's biggest strength was its structure and ability to organize information for linear thinkers. By 1993, Gopher had exhibited exponential growth in its user base. "In April of that year [1993], there were just over 460 registered gopher servers; by December there were over 4800".[10] Soon, the University of Minnesota's computing center was forced to accept Gopher as it took off around the country's Universities. This was the system that could have been the new paradigm. It had a jump start on the WWW with a substantial user base. However, the Gopher team would soon make a big mistake.

Companies were going to the University and wanted to use Gopher for their systems. So, the university decided to start charging everyone except for non-profit and academic institutions. Even though the university was not charging everyone, they still crossed the line between General Public Domain (GPD) and a licensed product. "It pissed a lot of people off," remembers McCahill. "It was a funny time. There was a recession on and the school just didn't want to give stuff away for free."[11] The gopher team got flamed in a monumental way. In 1994, Gopher lost its ground relatively quickly and eventually got consumed by the WWW.

Tim Berners-Lee's Vision


Tim Berners-Lee

Tim Berners-Lee was the son of two mathematicians who programmed some of the world's first computers. As Berners-Lee notes in his recent book, "They were full of excitement over the idea that, in principle, a person could program a computer to do most anything." He mentioned that he was watching his father write a speech for Basil de Ferranti (of Ferranti Computers, a seller of the early Mark I machine). His father was researching on the brain, thinking about the random associations that the brain makes and exploring the

[10] How the Web was Born, p. 140
[11] Mark McCahill phone interview.

possibility of computers behaving similarly. The idea struck Berners-Lee and he continued to think about that topic throughout his life.[12]

Berners-Lee continued in his education at Oxford, graduating in 1976 with a bachelor's in physics. After a few years at Plessey Telecommunications and DG Nash, Tim took a consulting job in 1980 with CERN, the European particle physics laboratory in Geneva. Here, Berners-Lee wrote his first attempt at information management, a program he called Enquire. He just wrote it in his spare time for personal use, to remember the connections at CERN among the various computers, projects and people at the lab. Berners-Lee credits this as the time that a larger vision took root in his head.

"Suppose all the information stored on computers everywhere were linked. Suppose I could program my computer to create a space in which anything could be linked to anything. All the bits of information in every computer at CERN, and on the planet, would be available to me and anyone else. There would be a single, global information space."

After only six months at CERN, Berners-Lee returned to England to work with John Poole, a former colleague. Poole had formed a company based on dot-matrix printer control software that Poole and Berners-Lee had worked on two years earlier. Berners-Lee helped enhance the software and bring it to market, including developing a special markup language for document preparation. However, in the spring of 1983, Berners-Lee needed a change from living in Britain, and applied for a fellowship back at CERN.[13]

Berners-Lee returned to CERN in 1984. By this time, the growth of networking and various protocols had continued to the point of sustaining various computers at CERN. As Berners-Lee mentions in his book, "Machines from IBM, DEC, Control Data--we had them all, as well as the new choice of PC or Mac in personal computers and different word processors." In this atmosphere, he wrote some RPC (remote procedure call) tools.

Given Berners-Lee's experience with designing Enquire and Tangle, and after reflecting upon the need for a new documentation system at CERN, he was able to

---

[12] Weaving the Web, pgs. 3-4
[13] Weaving the Web, pgs. 11-12

formulate a vision of a new system that would become the World Wide Web. The grand design for the system was for it to be a universal information space - a system that could store anything and the links between pieces of information.[14] Berners-Lee decided on a web-like structure using hypertext, in which nodes of information were connected by links, which could connect any two pieces of information together to describe a relationship between those pieces of information. By completely abandoning any notion of constrained structure, and by using simple protocols and formats, Berners-Lee created a system that could be acceptable to all - "a system with as few rules as possible."[15]

To represent the nodes in this web of information, Berners-Lee chose a simple hypertext document format, represented by standard ASCII text documents. Special spans of text in each document, such as headings and links, were set off from the rest of the text using simple tags like <BODY></BODY>. The most important aspect of this representation of hypertext is that it is extremely platform-neutral, as plain text can be read by every computer platform. Berners-Lee realized this, and used the cross-platform nature of ASCII text to truly exploit the diversity of different computer systems and networks that could possibly use the Web.[16]

Following from the establishment of this arbitrary web structure for his universal information space, Berners-Lee made the key insight that since any node could be linked to another node, there existed an equivalency between all nodes from a systems point of view. A page buried deep inside a proprietary help system would be no different from a transcript typed into a flat text file -- each node would need to have a unique address, allowing a viewer to bridge across information systems seamlessly within a single viewer. In addition to unifying different data formats, this common addressing scheme sought to unify various hardware platforms as well, since the address would be platform-neutral.[17]

CERN's organizational structure dictated another important aspect of this new system. Since CERN's scientists were constantly coming and going, and doing work

[14] Weaving the Web, pg. 16
[15] Weaving the Web, pg. 15
[16] Weaving the Web, pg. 15
[17] Weaving the Web, pg. 16

from remote locations[18], it was essential that the new information system be network-aware. This measure strengthened the platform-independence of the system, as common networking protocols are often used to connect disparate systems together. TCP/IP was chosen because it was used already by the Unix world, and VAX machines could be patched to support TCP/IP as well.[19] However, perhaps the most important and desirable feature of a network-distributed system is that new systems can come online without disrupting or modifying any existing systems, allowing the system as a whole to scale with the size of the network.[20]

Scalability was a key issue that Berners-Lee was aware of from the beginning. It proved to be the downfall of many more robust systems attempting to solve the information management crisis. He realized that any one central control point for any aspect of the system would limit its ability to scale up to thousands or millions of users/servers. It could also limit access to certain groups of people.[21] A key aspect of decentralization was the realization that a central link database would fatally cripple the system's ability to scale. The benefits of a central link database is that there would never be a broken link; however, Berners-Lee decided that the ability to scale up was worth the occasional inconvenience of broken links.[22]

Collaboration, and more generally, online editing of hypertext documents was a persistent theme in Berners-Lee's vision for a universal information space. On the most fundamental level, users would be able to add new links from one node to another as they browsed.[23] This simple notion would allow users to keep the information stored in the link structure of the web up to date as they browsed, and it would be easy to link new content into an existing web. In the Information Management proposal, Berners-Lee also proposed the need for the management of private links from each node, so that users could mark up a node with their own comments in the form of links that would not be seen by others.[24]

Berners-Lee elucidated even further on his vision of collaboration in 1996 in an

[18] Information Management: A Proposal, pg. 10
[19] Weaving the Web, pg. 19
[20] Weaving the Web, pg. 16
[21] Weaving the Web, pg. 16
[22] Weaving the Web, pg. 16
[23] Weaving the Web, pg. 16

interview published in the MIT Technology Review. He envisioned web documents as fully collaborative living documents, able to be edited in real-time by several people simultaneously and include live video and/or audio feeds from each participant:

> BERNERS-LEE: Say that you conduct a meeting as a hypertext document. You start by dragging in a video version of yourself, with real-time sound. You remind those invited to come by sending them a hypertext e-mail with a pointer to the meeting. To join, they just follow the link. They can not only read this meeting/document, but they also write to it. Some join by audio and some drag their own video into the document. People introduce points by writing them into the minutes, making links to background material. At one point in the meeting three people realize they need to discuss something separately, and with a single keystroke one forks off a new meeting document that they will catch up with later.[25]

Clearly, Berners-Lee envisioned a large degree of two-way interaction on the Web, especially for the purpose of facilitating group communication.

Above all, Berners-Lee desired that a simple, straightforward design with lots of room for future expansion. Too much innovation and new concepts all at once would result in lower acceptance for his vision. Berners-Lee's decision to keep his vision simple and pragmatic allowed others to quickly adopt the Web into their own software efforts.

From Vision to Proposal

Berners-Lee wished to implement his vision as a new information management system for CERN. Mike Sendall, Berners-Lee’s boss, asked him to write a proposal for the system, which was entitled “Information Management: A Proposal”. This proposal encompassed most of Berners-Lee's vision for a universal information space, and it especially accentuated the aspects that were most important for the scientists and management at CERN. A large section of the proposal is dedicated to explaining how traditional hierarchical structures like those used in older information systems at CERN were inflexible and therefore forced unnecessary constraints onto information stored within these systems. Berners-Lee observed that organizations and relationships at CERN often did not follow rigid hierarchical structures, especially when looked at over time. Information about a project should be able to grow and evolve along with the

---

[24] Information Management: A Proposal, pg. 10

organizations that work on it, and therefore "the method of storage must not place its own restraints on the information".[26]

The proposal also touched upon several other issues important to CERN, including specific examples of existing network-based services that would be converted over to Web services, such as the phone book and various help systems. Berners-Lee also mentioned the possibility of making the system work over DECnet, as CERN's computer population had a significant percentage of DEC VAX mainframes, which primarily used DECnet. And finally, the proposal briefly included an overview of the project management of developing the system, which was estimated at 2 people's time for 6-12 months, and then further effort to get CERN's existing systems online in the new system.

The Engineering Realities

Having created a vision of a universal information space that allows both viewing and cooperative editing of hypertext documents, Tim Berners-Lee faced the challenge of turning his ambitious vision into an engineering reality. Berners-Lee's first thought was to add networking capabilities to existing hypertext systems. At the European Conference on Hypertext Technology in 1990, Berners-Lee approached Ian Richie of OWL Ltd. about merging his ideas with OWL's Guide hypertext system. The Guide was "astonishingly like what [Berners-Lee] had envisioned for a Web browser," including the functionality to edit hypertext documents.[27] However, Berners-Lee was unable to capture Ritchie's interest. Ian Richie, the founder of OWL, had a different view. OWL's Guidex already had most of the features, including retrieval of documents over a network, that Berners-Lee wanted for the Web. What OWL did not realize was the fact that the networking aspect of an information system would prove to be more important than its hypertext aspects. Dynatext, created by Electronic Book Technology, was another potential hypertext base for Tim Berners-Lee's Web, but here too, he was unable to make

[25] Technology Review June 1996
[26] Information Management: A Proposal, pg. 4
[27] Weaving the Web, pg. 26

his case. Berners-Lee attributed the lack of enthusiasm for his ideas in the hypertext community partly to the lack of a good demonstration of the Web. He felt that "explaining the vision of the Web to people was exceedingly difficult without a Web browser in hand".[28] It was becoming clear that if the Web were ever to see the light of day, Tim Berners-Lee would have to design and build it himself, from scratch.

The first implementation of the Web was completed in about two months by Tim Berners-Lee, working alone. Berners-Lee had developed a graphical browser/editor that could read and write HTML documents through HTTP. He also developed the first Web server at *info.cern.ch* to serve up the documentation and his notes for HTTP and HTML.[29] Early HTML and HTTP were extremely simple out of necessity. Being the only developer, Berners-Lee could not afford to implement a complex set of functionalities. The HTTP protocol is therefore layered on top of existing Internet standards such as TCP/IP (Original HTTP, 1991). For example, when a URL such as http://web.mit.edu/index.html is entered into a Web browsers, the HTTP connection takes place in three phases. First, the type of the URL request is determined to be HTTP. Then the name of the remote server, *web.mit.edu*, is extracted from the URL and the actual IP address is looked up via the DNS system. Since no TCP port number on *web.mit.edu* has been specified, the default port for HTTP connections (port 80) is used. The Web browser now makes an ordinary TCP connection to port 80 on the computer named *web.mit.edu*. Once the connection is established, the Web browser sends a single line of ASCII characters, which is the actual HTTP request. In our example the line will read "GET index.html." The HTTP server running on *web.mit.edu* receives this request and tries to find the file "index.html." If the file exists and is readable by all users, then it is sent to the Web browser. But if for some reason the file cannot be located or cannot be read for some other reason, an error message is sent to the Web browser instead. When the server has finished sending either the file or the error message, it closes the TCP connection with the Web browser, thus ending the HTTP connection. It is then up to the Web browser to correctly display the document that it has received.

As we can see from the above example, the original HTTP design was a "non-

---

[28] Weaving the Web, pg. 27
[29] Weaving the Web, pg. 29

computer-scientist's protocol" that favored simplicity over efficiency.[30] Its most important feature, however, was that it worked *well enough* to be a usable system. Similarly for HTML, simplicity and functionality were its strengths. The original HTML specification contained little more than a dozen distinct markup tags. Only the most rudimentary document markups were supported: paragraphs, lists, highlighting of text, headings, indices, glossaries, and of course hypertext links. In comparison, HTML 4.01 defines 91 distinct tags and literally hundreds of additional attributes that can be used to modify the behavior of these tags. All of the tags in HTML 4.01 are useful today, but if Berners-Lee had made the very first version of HTML just like HTML 4.01, chances are the Web would not have taken off at all.

The simplicity of HTML and HTTP also had their drawbacks, however. One of the features Tim Berners-Lee had to sacrifice was the idea of cooperative editing of HTML pages over the Web. In order to implement such a two-way Web, the HTTP protocol would have needed a good authentication system so that users can be given permission to write HTML documents onto a remote computer. The 1992 version of HTTP shows that some provision was made for such an authentication system. In the HTTP request specification, one of the fields is named "Authorization," and its purpose was to allow the originator of the request to authenticate himself or herself to the HTTP server. Once a user is authenticated, he or she can be given permission to changed HTML documents on the server using the "POST" and "PUT" methods. "POST" and "PUT" are similar to the "GET" method, and they allow the user to amend and create files on the server, respectively. These features, if they were implemented, could have supported Berners-Lee's vision of a two-way Web. But the "Authorization" field was designated at To Be Specified (TBS) in 1992, and even today, there is still no simple way to directly modify existing HTML pages on the Web. It is interesting to note that the W3C's Amaya project is still striving for Berners-Lee's ideal of a two-way Web by integrating browsing and editing functionalities into a single Web client. However, for

[30] Dave Gifford interview.
[31] Pei Wei email interview.
[32] Pei Wei email interview.
[33] Thomas Bruce interview.
[34] Thomas Bruce interview.
[35] An Interview with Tim Berners-Lee (video).

the foreseeable future, most HTML documents on the Web will remain read-only.

Despite the shortcomings of the first implementation of the Web, Berners-Lee remained steadfast on his altruistic goal of building a universal information space for the entire world. In an email to Dan Connolly on November 8, 1991, Berners-Lee summarized the vision of the Web saying, “I hope I have clarified the W3 team's philosophy, and perhaps convinced you to contribute, to our mutual (and the world's) benefit.”[36]

The Struggle For Acceptance

When Tim Berners-Lee implemented the first version of the Web, the world did not beat a path to his door. Just as Thomas Kuhn predicted that new scientific paradigms are met with much resistance, the World Wide Web met its share of resistance. Two important sources of early resistance to the Web were CERN and the hypertext community. It is a common misconception, perpetuated to a degree by Tim Berners-Lee’s own accounts, that neither CERN management nor the hypertext community saw the usefulness of the World Wide Web. It is true that they underestimated its importance, as almost everyone did at the time. But CERN and the hypertext community were not blind to the potentials of a networked information system. Then rejected Tim Berners-Lee’s project, which is quite different from rejecting the premises of the World Wide Web. We will explore some of the reasons for CERN and the hypertext community to reject the early Web.

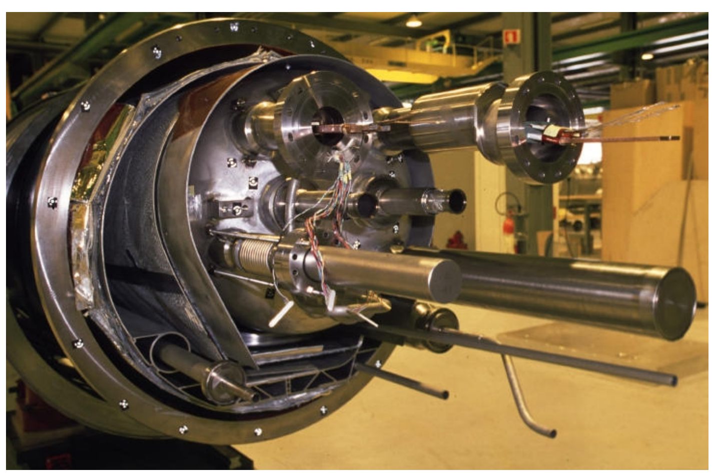


Figure 6 Particle accelerator at CERN.

CERN

As we have said earlier, CERN is the center of European high-energy physics

[36] Email from Berners-Lee to Dan Connolly, included as Item 1 in Appendix A.

research. As such, it had a tremendously diverse culture and a great deal of complex social needs. For example, the people working at CERN were not all working *at* CERN. Many of the scientists involved in CERN projects were scattered all over Europe, the United States and even Japan. There were also a large number of projects going on at the same time. Such a heterogeneous environment cried out for a networked information system. Tim Berners-Lee, as a software consultant for CERN, was keenly aware of this need. In his numerous proposals to CERN management, Tim Berners-Lee always stressed the usefulness of his World Wide Web project to the people at CERN. CERN was of course also well aware of its own needs, and it seems puzzling that they did not at least explore Tim Berners-Lee's ideas more fully.

However, the puzzle is much less puzzling when we consider the fact that CERN's primary focus is physics research. CERN's policy regarding software systems is "buy, not build."[37] However, this policy alone is not the only reason CERN did not take up Tim Berners-Lee's World Wide Web project. According to David Williams, Tim Berners-Lee focused a lot of his persuasive efforts on the hypertext aspects of the Web. However, there were a number of people at CERN, Williams included, who did not feel that hypertext was an appropriate way to represent information. Williams also thought, "Explaining his ideas is not [Tim Berners-Lee's] real strength."[38] As we can see from FIGURE 7, Tim Berners-Lee's diagram of how the Web works,[40] presents quite a complex picture of his ideas. This complex picture presents a direct contrast to the relatively simple implementation of the Web. This disparity between the simplicity of the Web's implementation and the complexity of Berners-Lee's explanations may explain why CERN hesitated to back a full-scale project to develop the Web. As a physics research institution, CERN had to devote most of its resources to that end. However, if Tim Berners-Lee had presented the Web as a less daunting project, it is possible that CERN might have undertaken to support his efforts more.

[37] Weaving the Web
[38] David Williams email interview.
[39] Information Management: A Proposal
[40] Information Management: A Proposal

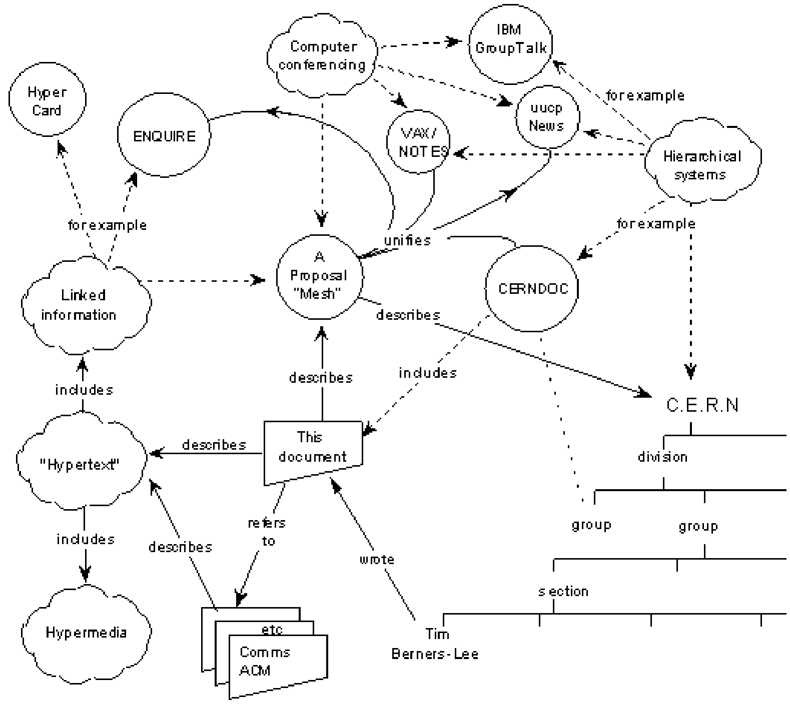


Figure 7 Tim Berners-Lee's flow chart for the Web

The Hypertext Community

In trying to minimize the development efforts required for the World Wide Web, Tim Berners-Lee had approached several hypertext system builders about possible collaboration. Berners-Lee believed that once people saw his simple implementation of the Web, they would realize the enormous potential of a global information space. However, his early implementation of the Web was not an impressive system to people within the hypertext community. At the Hypertext '91 conference in San Antonio, Tim Berners-Lee demonstrated the WorldWideWeb browser on his NeXT machine to the attendees. However, as a hypertext system, HTML was far too simple to make for a convincing demo.[41] The key to Berners-Lee's idea was of course the marriage of hypertext and the Internet. But to the hypertext community, his implementation was not an interesting *hypertext* system because it addressed none of the issues facing hypertext systems of the time.

For example, link consistency was a big concern for most hypertext systems in the early ‘90s. When contents move in a system, links that point to the contents can become broken. Many hypertext systems were designed to deal with this problem. The Web on the other hand, completely ignored the problem of link consistency. Tim Berners-Lee was aware of the link consistency problem, but he correctly recognized that there was no simple, scalable way to ensure global link consistency in a worldwide information system. However, by ignoring the interesting hypertext research problems, Berners-Lee was not able to persuade the hypertext community to help him develop the Web.

Why the Web Won

Despite the resistance Tim Berners-Lee faced at CERN and in the hypertext community, the Web did beat out its competitors to become the most widely used electronic information system. The triumph of the Web comes from a combination of many factors. Tim Berners-Lee's promising vision, the simplicity and openness of the Web's design and initial implementation, the tremendous grass-roots support from users and developers, and fortuitous timing all contributed to the eventual success of the Web as a new paradigm.

Tim Berners-Lee, though he may have erred in presenting too grand a vision to CERN, received an entirely different reaction from the world at large. For people outside CERN, the idea of a global information space was very appealing. However, the simplification of Tim Berners-Lee's vision was the only way for the Web to take roots and grow. The simplicity of early HTML and HTTP not only made them easy to implement but also easy to extend incrementally. Unlike more mature hypertext systems such as Dynatext, which had consistency checks to ensure their hypertext documents were valid, HTML was extremely lax in terms of error checking. While it created something of a problem because ill-formed documents could not be viewed, the lack of error checking was the most important factor in the growth of the Web. Since there is no real way to ensure that an HTML document conformed to the specifications given by Berners-Lee, browser implementers were free to add their own tags to HTML and have them be appropriately interpreted by their own browsers. All a browser implementer had to do was let his users know that a new tag was supported by his browser. For example

[41] Peter Brown phone interview.

the developers at NCSA added tags to display graphics to their browser implementation in just such a unilateral fashion.

When Tim Berners-Lee released his creation to the people, the Web enjoyed tremendous grass-roots support on the Internet. Berners-Lee had always known the importance of gaining a "critical mass" of users and information on the Web. In fact, the first Web server that he set up at *info.cern.ch* served documentation on the Web itself and instructed people on how they can set up their own Web servers.

Here too, the simplicity of HTTP and HTML were of great value. Early HTTP was so simple that an HTTP server could be implemented as a UNIX shell script. Similarly, the HTML defined by Berners-Lee was simple enough that several individuals were able to develop Web browsers from scratch. Pei Wei, then at the University of California at Berkeley, developed Viola, a graphical browser for the X Windows environment (Figure 8). Wei commented on the "simple and elegant" URL scheme used by HTTP that attracted him to the Web.[43] Wei believes that if HTTP and HTML were not "plainly easy for a developer to work with," then the Web would have lost that developer and "that much of a resource for this grass-root supported system".[44] Other developers that created early browsers such as Tom Bruce, whose Cello browser was the earliest Windows based browser, also credit the simplicity of the Web for its quick acceptance. Bruce believes that Berners-Lee was able to "avoid letting the perfect become the enemy of the good" by keeping early HTML and HTTP simple and adding

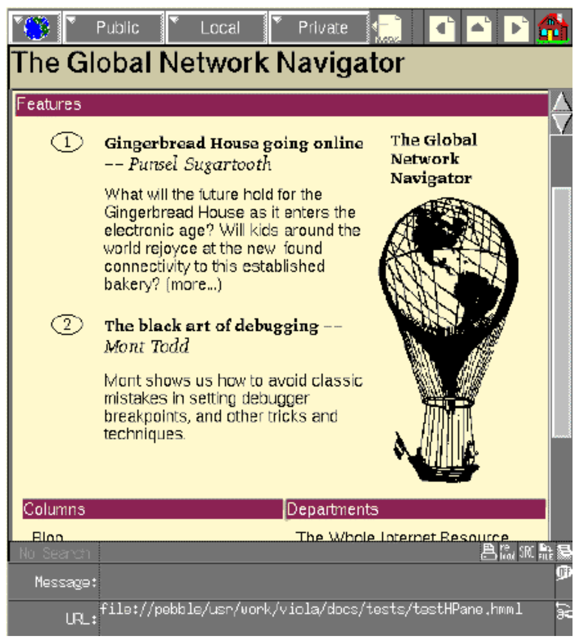


Figure 8 ViolaWWW Screenshot.

[42] Mark McCahill phone interview.
[43] Pei Wei email interview.
[44] Pei Wei email interview.

more functionality as the Web grew.[45] However, the Windows platform was largely ignored by the early Web enthusiasts. Even Berners-Lee's cross platform library, libwww, worked poorly under Windows because of memory allocation issues. Thomas Bruce, in the course of writing Cello, rewrote the entire libwww for Windows.[46] For a lone developer, this was a great deal of work. But because the Web was a simple enough and exciting enough, it spread to all the important platforms through the work of its grass-roots supporters.

Once free Web browsers became available on all platforms and independent HTTP servers began to come online, the Web took off quickly. Tim Berners-Lee observed that since 1991, the Web has been growing at an exponential pace.[47] The advent of cheap PCs and a commercialization of the Internet further fueled its incredible growth. In less than 10 years, the Web went from a system no one wanted to a household word thanks in large part to the grass-roots support it enjoyed in its early days.

The World Wide Web had some additional advantages over its closest competitor, Gopher. In the Spring 1992 issue of the *Electronic Networking* journal, Berners-Lee wrote that since Gopher uses the "directory and file model to implement a global information system," it would "map into the Web very naturally, as each directory (menu) is represented by a list of text elements linked to other directories or files (documents)."[49] In other words, the Web was a more general system that is capable of encompassing most of Gopher's capabilities. Another advantage the Web had over Gopher was that the "web gave information away and had ads to support it."[52] Companies saw the opportunity to make money and therefore chose the web over Gopher. The Web was "very pretty" and while librarians loved Gopher for its orderliness, companies loved the Web for its graphics capabilities.

While the graphical capabilities of Mosaic did much to popularize the Web, such

[45] Thomas Bruce interview.
[46] Thomas Bruce interview.
[47] An Interview with Tim Berners-Lee (video).

[49] Electronic Networking, pg. 56

[51] Mark McCahill phone interview.
[52] Mark McCahill phone interview
[53] Thomas Bruce phone interview.

proprietary extensions were a double-edged sword. On the one hand, they cause incompatibilities among different browser implementations, thus dividing Berners-Lee's vision of a universal information space. An HTML document containing Mosaic specific tags would not display correctly on any browser except NCSA Mosaic. On the other hand, proprietary extensions were quick and direct ways for enterprising developers to add new capabilities to the Web. In the early days of the Web, the benefits of these extensions outweighed the drawbacks. The graphical capabilities of Mosaic were one of the main reasons the Web became so attractive to people outside the circles of academics and research. As the saying goes, "graphics sells," and in the case of the Web, the graphics gave it a tremendous leg up on competing systems such as Gopher.[54] However, as the Web grew and Mosaic became the dominant browser, the danger that a group of aggressive developers could hijack the development of the Web became increasingly present. It was clear that in order for Berners-Lee's vision of the universal Web to survive, some neutral body was needed to build consensus among the various forces that were pulling the Web in different directions.

Adoption of the New Paradigm

The Web needed to leave CERN for it to truly succeed - it was, after all, a particle research facility and not focused on computer science research. Berners-Lee therefore moved his work and advocacy for the Web to the Massachusetts Institute of Technology's Laboratory for Computer Science and started the World Wide Wed Consortium (W3C). "The W3C was founded in October 1994 to lead the World Wide Web to its full potential by developing common protocols that promote its evolution and ensure its interoperability."[55] Michael Dertouzos, the Director of the Laboratory for Computer Science describes that there was "tremendous synergy" between him and Berners-Lee when they met in Zurich in 1994. Dertouzos was interested in creating an "information marketplace" and saw the world-wide web as the "potential underlying mechanism" for

54 Mark McCahill phone interview.
55 http://www.w3c.org web page.

his vision.[56] MIT had just successfully spun off the X Consortium and this provided good timing for Berners-Lee to set up the W3C when he was invited to do so by Dertouzos. He saw it as his opportunity to continue to drive the future of the Web as it grew. CERN allowed the Web to leave and gave up its rights to the technology that Berners-Lee had developed there. David Williams, a manager at CERN, credits this happening to Berners-Lee and Robert Cailliau. He says, "Tim was strongly of the opinion that only an open release would allow it [the Web] to take-off. While we are all happy with what happened I tend to feel that CERN should have tried to get a little more public recognition for our/his work."[57] But, CERN did not try to do that, and the Web took off because of it.

The W3C is very active today in promoting standards for different web technologies. Companies pay a fee for membership and, while they are not forced to abide by the recommend standards that are developed by the W3C, they are incompatible with others if they chose not to. The aforementioned "browser wars" in which Mosaic and other browsers battled to outdo each other by adding more and more proprietary features was brought under control by the consensus-building approach of the W3C. This stabilization and standardization fostered the growth of the Web, as developers and companies can depend on there not being major differences and incompatibilities between the different browsers.

The birth of the W3C signifies the acceptance of the Web as a new paradigm in information sharing. The W3C is a forum for the building of consensus on the direction of the Web. It also represents the community that lives and works within this engineering paradigm. By working to extend the Web and fulfill its potential, the work of the W3C represents the kind of normal engineering that goes on within an established paradigm. Once people began to accept the W3C as an authority in matters regarding the Web, the engineering revolution has truly come full circle.

---

[56] Michael Dertouzos interview.
[57] David Williams email interview.

Conclusion

We have tried to tell the story of the Web in terms of an engineering revolution. The progression from the existing paradigm consisting of unconnected computers and isolated islands of information to a connected world of universal information sharing is truly a revolutionary development. We have shown the similarities and differences between Kuhn's model of scientific revolutions and engineering revolutions.

People today do not give a second thought about looking for information on the Web. While it may not always be obvious as to how to find the information, it's almost guaranteed that it exists on some Web page. People just 10 years ago looked for their information in a dramatically different way: the world has truly experienced a paradigm shift.

Bibliography

Books

Berners-Lee, Tim, Weaving the Web: The Original Design and Ultimate Destiny of the World Wide Web by Its Inventor. HarperCollins 1999

Gilles, James, and Cailliau, Robert, How the Web was Born. Oxford University Press, 2000

Naughton, John. A Brief History of the Future: The Origins of the Internet. Weidenfeld: London, 1999.

Papers and Journal Articles

Berners-Lee, T.J., Cailliau, R. and Groff, J.-F. "The World Wide Web." Computer and Networks and ISDN Systems 25. 1992: pp454-59.

B. Kahle et al., "Wide Area Information Servers : An Executive Information System for Unstructured Files", in: *Electronic Networking: Research, Applications and Policy,* Vol. 2, No. 1 (Meckler, New York, 1992) 59-68.

Berners-Lee, T.J. et al., "World-wide web: the information universe", in: *Electronic Networking: Research, Applications and Policy,* Vol. 2, No. 1 (Meckler, New York, 1992) 52-58.

Berners-Lee, T.J. et al., "World-wide web: an information infrastructure for high-energy physics", in: D. Perret-Gallix, ed., *Proc. International Workshop on Software Engineering and Artificial Intelligence for High Energy Physics,*La Londe, France, January 1992.

McCahill, M. et al., "The Internet Gopher: An Information Sheet", in: *Electronic Networking: Research, Applications and Policy,* Vol. 2, No. 1 (Meckler, New York, 1992) 67-71.

Berners-Lee, Tim. "Information Management: A Proposal." CERN, 1989.

Berners-Lee, Tim and Cailliau, Robert. "World Wide Web: Proposal for a HyperText Project." CERN, 1990.

"The Web Maestro: An Interview with Tim Berners-Lee", Technology Review, July 1996, archived at http://www.techreview.com/articles/july96/bernerslee.html

Roush, Wade, "MIT Reporter: Spinning a Better Web", Technology Review, April 1995, archived at http://www.techreview.com/articles/apr95/BernersInterview.html

Interviews

Email interview with David Williams, Computing and Networks Division CERN, November 11, 2000

Interview with Alan Kotok, Associate Chairman of the World Wide Web Consortium, November 13, 2000

Telephone interview with Mark McCahill, Computer and Information Services, University of Minnesota, November 15, 2000

Email interview with Thomas Bruce, Legal Information Institute at Cornell Law School, November 15, 2000

Interview with Dave Gifford, Programming Systems Research Group, MIT Lab for Computer Science, November 15, 2000

Telephone interview with Peter Brown, University of Kent, November 17, 2000

Telephone interview with Ian Ritchie, Office Workstations Limited (OWL), November 20, 2000

Email interview with Pei Wei, University of California Berkeley/O'Reilly & Associates, November 21, 2000

Telephone interview with Doug Engelbart, Stanford Research Institute/Bootstrap Institute, November 27, 2000

Interview with Michael Dertouzos, Directory of MIT Lab for Computer Science, November, 28, 2000

Videos

A conversation with the man who invented the World Wide Web [videorecording] : Tim Berner-Lee [sic], 1996, MIT Archives collection

Demonstration of NLS, Doug Engelbart, http://sloan.stanford.edu/MouseSite/1968Demo.html

Web Sites

RFC1630: Universal Resource Identifiers in WWW
Http://www.ietf.org/rfc/rfc1630.txt?number=1630

Archives of the www-talk forum
Http://ksi.cpsc.ucalgary.ca/archives/WWW-TALK/archives.html

List of Original HTML Tags, 1991.
http://www.w3.org

Berners-Lee, Tim. HyperText Transfer Protocol Design Issues, 1991.
http://www.w3.org

Grahn, J. Harmon. "The World Wide Web."
http://www.olypen.com/harmon/np304.htm

Appendix A: Selected Emails from the WWW-Talk archive

All of these emails are hosted at http://ksi.cpsc.ucalgary.ca/archives/WWW-TALK/archives.html.

Item 1 – Email from Tim Berners-Lee to Dan Connolly

Re: Motif browser status

timbl (Tim Berners-Lee)

```
Date: Fri, 8 Nov 91 13:35:26 GMT+0100
From: timbl (Tim Berners-Lee)
To: connolly@pixel.convex.com
Subject: Re: Motif browser status
Cc: kharris@pixel.convex.com, www-talk

Dan,

Thanks for your message.  Obviously you know what you are doing with
X11 browsers - we are impressed by what you have done to date. I was
interested to hear that you are working on AVS - I have had some
contact with AVS people at UNC.

You make a good point that the world has been waiting for a good
formatted text widget under Motif. One exists under NeXTStep, Robert
Cailliau is just adapting one for the Mac for hypertext, but under
Motif it has been lacking.  Of course, hundreds of people have
written them: all the word processors have them in, and products like
dynaText, etc. However, there is none in the public domain.

CERN like Convex has a copyright on all code, but we are doing our
best to release W3 code as widely as possible, and possibly overcome
this limitation. Why?

The concept of the web is of universal readership. If you publish a
document on the web, it is important that anyone who has access to it
can read it and link to it. In order to make this possible, we don't
need very new technology -- what we do need is

        1.      A common open naming/addressing format
        2.      Sufficiently powerful underlying protocols
        3.      Sufficiently powerful data formats
        4.      Some free implementations

Now we have defined the (1), which did not exist before. We have
supplemented the (2), where some protocols do exist. We have added a
little to (3) though we will use all existing and new formats. We
have written some code.

You say your work would be of considerable valuer to convex. Yes,
that is true. You must ask yourself whether it would be of more value
to convex if kept private or released for general consumption. If you
release it,
```

```
  -     Convex gets the credit and a higher profile,
          (as Thinking Machines has with WAIS indexers for example).

  -     Anyone in the world can read the information you supply
          with the same tool as they use for other information.

  -     You get a lot of useful feedback from users on the network

  -     A lot of people would be able to profit from what you have
         done
```

You have to compare this scenario with that if you keep the code private. You will be able to use it internally. Would convex be able to profit from by selling it? If so, how many people would actually buy it? Will the AVS project benefit from a closed private documentation scheme?

On these grounds alone, you may conclude that it is in Convex's interest to release the code. Still, you ask what we can "put on the table". If it would make it easier to justify the release of code, we would be happy to make all CERN-developed W3 code officially available to Convex under a more or less formal joint project agreement. Note that we are producing a parallel set of parsers and access mechanisms for HTML, newgroups, WAIS, prospero, etc. We have gateways, and other browsers. The line-mode browser you know, the Mac one is coming along, we may have a full-screen character grid browser too. We are currently unifying the browser architecture so that all access mechanisms can be used by all browsers. I'm not sure that either of our sides would want to be contractually bound to produce or maintain anything - the agreement would be just as-is code sharing of what exists when it exists, no strings.

You ask about graphics. That cannot be our next priority, as we need to get the new architecure and general format negociation worked out. In many cases, we find that there are GIF/TIFF viewers on various platforms, and one can link in to them. We don't want to make a new graphics file format a la Mac/PICT, but we are intrerested in conversion code. Have you heard of editable Postscript? That might be what you are looking for. (See http://info.cern.ch/hypertext/Standards/PostScript/IPF.html)

I don't know whether your company has a mechanism for allowing code to be released into the public domain (or General Public License). If it is politically impossible, then that's a pity. (We do have a group of students in Finland working on an X implementation, and if that doesn't work out we could write it ourselves. It may also be that more that one implementation with a different style will be interesting. Obviously it would be rather a duplication of effort, though we are under a lot of pressure from our management and users to put this at the top of the agenda.)

I hope I have clarified the W3 team's philosophy, and perhaps convinced you to contribute, to our mutual (and the world's) benefit.

Tim

PS: Yes, I think you ought to be on www-talk, Dan. I'll put you on.

```
The traffic is not too high.
_____________________________________________________________
Tim Berners-Lee                       timbl@info.cern.ch
World Wide Web project                (NeXTMail is ok)
CERN                                  Tel: +41(22)767 3755
1211 Geneva 23, Switzerland           Fax: +41(22)767 7155
```