CHAPTER 3

# info.cern.ch

While it seemed to be uphill work convincing anyone at CERN that global hypertext was exciting, one person was an immediate convert: Robert Cailliau.

Though now the Electronics and Computing for Physics division, by coincidence Robert had in 1980 been in the same Proton Synchotron division as I, and had in fact written the text-formatting program I had used to print the Enquire manual. A Flemish-speaking Belgian, Robert had had the lifelong frustration of people insisting on addressing him in French. After taking an engineering degree at the University of Ghent he picked up a master's at the University of Michigan, an experience that left him with an accent in English that is impossible to identify. Indeed, it became a parlor game for newcomers at CERN to try to guess exactly where he was from.

A dapper dresser who methodically schedules haircuts according to the solstice and equinox, Robert is fastidious in all

things. He is the kind of engineer who can be driven mad by the incompatibility of power plugs. No wonder, then, that he would be attracted to a solution to computer incompatibility, especially coming with a simple user interface. In the marriage of hypertext and the Internet, Robert was best man.

Robert's real gift was enthusiasm, translated into a genius for spreading the gospel. While I sat down to begin to write the Web's code, Robert, whose office was a several-minute walk away, put his energy into making the WWW project happen at CERN. He rewrote a new proposal in terms he felt would have more effect. A CERN veteran since 1973, he lobbied among his wide network of friends throughout the organization. He looked for student helpers, money, machines, and office space.

By the time Mike Sendall approved my purchase of the NeXT machine, I had already gone to the hypertext industry looking for products onto which we could piggyback the Web. At CERN there was a "Buy, don't build" credo about acquiring new technology. There were several commercial hypertext editors, and I thought we could just add some Internet code so the hypertext documents could be sent over the Internet. I thought the companies engaged in the then fringe field of hypertext products would immediately grasp the possibilities of the Web. Unfortunately, their reaction was quite the opposite. "Nope," they said. "Too complicated."

Undaunted, in September 1990 Robert and I went to the European Conference on Hypertext Technology (ECHT) at Versailles to pitch the idea. The conference exhibition was small, but there were a number of products on display, such as a multimedia training manual for repairing a car.

I approached Ian Ritchie and the folks from Owl Ltd., which had a product called Guide. In Peter Brown's original Guide work at the University of Southampton, when a user clicked on a hypertext link, the new document would be inserted right there in place. The version now commercialized by Owl looked astonishingly like what I had envisioned for a Web browser—the program





that would open and display documents, and preferably let people edit them, too. All that was missing was the Internet. *They've already done the difficult bit!* I thought, so I tried to persuade them to add an Internet connection. They were friendly enough, but they, too, were unconvinced.

I got the same response from others at the conference. It seemed that explaining the vision of the Web to people was exceedingly difficult without a Web browser in hand. People had to be able to grasp the Web in full, which meant imagining a whole world populated with Web sites and browsers. They had to sense the abstract information space that the Web could bring into being. It was a lot to ask.

The hypertext community may also have been slightly demoralized. Their small conference was not getting any bigger, and no one was sure where the field was headed. The lack of commercial successes had perhaps left a certain cynicism about bright new ideas that could change the world.

Another possibility I saw was called Dynatext, and was from Electronic Book Technology, a company in Rhode Island started by Andy Van Dam, the Brown University researcher who had coined the term *electronic book*. I thought the company's software could be turned into a Web browser/editor rather easily. However, like many hypertext products at the time, it was built around the idea that a book had to be "compiled" (like a computer program) to convert it from the form in which it was written to a form in which it could be displayed efficiently. Accustomed to this cumbersome multistep process, the EBT people could not take me seriously when I suggested that the original coded language could be sent across the Web and displayed instantly on the screen.

They also insisted on a central link database to ensure that there were no broken links. Their vision was limited to sending text that was fixed and consistent—in this case, whole books. I was looking at a living world of hypertext, in which all the pages would be constantly changing. It was a huge philosophical gap.

Letting go of that need for consistency was a crucial design step that would allow the Web to scale. But it simply wasn't the way things were done.

Despite the "Buy, don't build" credo, I came to the conclusion that I was going to have to create the Web on my own. In October 1990 I began writing code for the Web on my new computer. The NeXT interface was beautiful, smooth, and consistent. It had great flexibility, and other features that would not be seen on PCs till later, such as voice e-mail, and a built-in synthesizer. It also had software to create a hypertext program. Its failure to take over the industry, despite all these advantages, became for me a cautionary tale. NeXT required users to accept all these innovations at once—too much.

My first objective was to write the Web *client*—the program that would allow the creation, browsing, and editing of hypertext pages. It would look basically like a word processor, and the tools on the NeXT's system, called NeXTStep, were ideal for the task. I could create an application, menus, and windows easily, just dragging and dropping them into place with a mouse. The meat of it was creating the actual hypertext window. Here I had some coding to do, but I had a starting place, and soon had a fully functional word processor complete with multiple fonts, paragraph and character formatting, even a spellchecker! No delay of gratification here. Already I could see what the system would look like.

I still had to find a way to turn text into hypertext, though. This required being able to distinguish text that was a link from text that wasn't. I delved into the files that defined the internal workings of the text editor, and happily found a spare thirty-two-bit piece of memory, which the developers of NeXT had graciously left open for future use by tinkerers like me. I was able to use the spare space as a pointer from each span of text to the address for any hypertext link. With this, hypertext was easy. I was then able to rapidly write the code for the Hypertext Transfer Protocol (HTTP), the language computers would use to communicate over the Internet, and the Universal Resource Identifier (URI), the scheme for document addresses.





By mid-November I had a client program—a point-and-click browser/editor—which I just called *WorldWideWeb*. By December it was working with the Hypertext Markup Language (HTML) I had written, which describes how to format pages containing hypertext links. The browser would decode URIs, and let me read, write, or edit Web pages in HTML. It could browse the Web using HTTP, though it could save documents only into the local computer system, not over the Internet.

I also wrote the first Web *server*—the software that holds Web pages on a portion of a computer and allows others to access them. Like the first client, the server actually ran on my desktop NeXT machine. Though the server was formally known as nxoc01.cern.ch (NeXT, Online Controls, 1), I registered an alias for it—"info.cern.ch."—with the CERN computer system folks. That way, the server would not be tied by its address to my NeXT machine; if I ever moved its contents to another machine, all the hypertext links pointing to it could find it. I started the first global hypertext Web page, on the info.cern.ch server, with my own notes, specifications of HTTP, URI, and HTML, and all the project-related information.

At this point Robert bought his own NeXT machine and we reveled in being able to put our ideas into practice: communication through shared hypertext.

At long last I could demonstrate what the Web would look like. But it worked on only one platform, and an uncommon one at that—the NeXT. The HTTP server was also fairly crude. There was a long way to go, and we needed help. Ben Segal, who had a knack for adjusting staffing levels behind the scenes, spotted a young intern named Nicola Pellow. A math student from England, Nicola was working for a colleague in a neighboring building but didn't have enough to do.

A big incentive for putting a document on the Web was that anyone else in the world could find it. But who would bother to install a client if there wasn't exciting information already on the Web? Getting out of this chicken-and-egg situation was the task before us. We wanted to be able to say that if something was on the Web, then anyone could have access to it—not just anyone with a NeXT!

When I gave talks, I showed a diagram with machines of all types connected to the Internet, from mainframes with simple character-oriented terminals through PCs, Macs, and more. To make this possible, I urged Nicola to give the Web the best browser she could, but to assume as little as possible, so this interface could work on any kind of computer. The least common denominator we could assume among all different types of computers was that they all had some sort of keyboard input device, and they all could produce ASCII (plain text) characters. The browser would have to be so basic that it could even work on a paper Teletype. We therefore called it a *line-mode* browser, because Teletype machines and the earliest computer terminals operated by displaying text one line at a time.

Meanwhile, I took one quick step that would demonstrate the concept of the Web as a universal, all-encompassing space. I programmed the browser so it could follow links not only to files on HTTP servers, but also to Internet news articles and newsgroups. These were not transmitted in the Web's HTTP protocol, but in an Internet protocol called FTP (file transfer protocol). With this move, Internet newsgroups and articles were suddenly available as hypertext pages. In one fell swoop, a huge amount of the information that was already on the Internet was available on the Web.

The *WorldWideWeb* browser/editor was working on my machine and Robert's, communicating over the Internet with the info.cern.ch server by Christmas Day 1990.

As significant an event as this was, I wasn't that keyed up about it, only because my wife and I were expecting our first





child, due Christmas Eve. As fate would have it, she waited a few extra days. We drove to the hospital during a New Year's Eve storm and our daughter was born the next day. As amazing as it would be to see the Web develop, it would never compare to seeing the development of our child.

As the new year unfolded, Robert and I encouraged people in the Computing and Networking division to try the system. They didn't seem to see how it would be useful. This created a great tension among us about how to deploy our limited resources. Should we be evangelizing the Web? Should we develop it further on the NeXT? Should we reprogram it for the Mac or the PC or Unix, because even though the NeXT was an efficient machine, few other people had them? After all, what good was a "worldwide" web if there were only a few users? Should we tailor the Web to the high-energy physics community, so they'd have a tool that was theirs and would support it, since CERN was paying our salaries? Or should we generalize the Web and really address the global community, at the risk of being personally disenfranchised by CERN?

Trading in the NeXT for some ordinary computer would have been like trading in a favorite sports car for some truck. More important, the Web was already written for it. If we switched to developing the Web for the much more widely used PC, acceptance might be quicker, but the point was to get people to try what we already had. If we stopped progress and went back to redoing things for the PC, we might never get it done. I decided to stick with the NeXT.

As for the application, my gut told me I had to pursue my larger vision of creating a global system. My head reminded me, however, that to attract resources I also needed a good, visible reason to be doing this at CERN. I was not employed by CERN to create the Web. At any moment some higher-up could have questioned how I was spending my time, and while it was unusual to

stop people at CERN from following their own ideas, my informal project could have been ended. However, it was too soon to try to sell the Web as the ultimate documentation system that would allow all of CERN's documents, within and between projects, to be linked and accessible, especially given the history of so many failed documentation systems. Small but quantifiable steps seemed in order. Our first target, humble beginning that it was, would be the CERN telephone book.

The phone book existed as a database on CERN's aging mainframe. Bernd Pollermann, who maintained it and all sorts of other central information, was charged with somehow providing all this material to each and every user on his or her favorite system. I managed to persuade Bernd that the Web was just what he needed to make life a great deal simpler. If he created a server, I told him, we would get the browsers onto everyone's desktop. He went for it.

I got my simple server to run on the mainframe, then chopped it in two, so that the essential HTTP-related Internet functions were done by my code (written in C language) and Bernd was left to write the rest of the server in his favorite language, "REXX." To make all the documents available, he just had to learn to write HTML, which took him only a few afternoons. Soon the entire world of his search engines, databases, and catalogues was available as hypertext.

That brought us back to the search for a browser. We started porting Nicola's line-mode client onto all sorts of machines, from mainframes through Unix workstations to plain DOS for the PC. These were not great showcases for what the Web should look like, but we established that no matter what machine someone was on, he would have access to the Web. This was a big step, but it was achieved at some sacrifice in that we decided not to take the time to develop the line-mode browser as an editor. Simply being able to read documents was good enough to bootstrap the process. It justified Bernd's time in getting his servers up. But





it left people thinking of the Web as a medium in which a few published and most browsed. My vision was a system in which sharing what you knew or thought should be as easy as learning what someone else knew.

Mundane as it was, this first presentation of the Web was, in a curious way, a killer application. Many people had workstations, with one window permanently logged on to the mainframe just to be able look up phone numbers. We showed our new system around CERN and people accepted it, though most of them didn't understand why a simple ad hoc program for getting phone numbers wouldn't have done just as well.

Of course, we didn't want our brainchild with all its tremendous potential to be locked in at this rather pedestrian level. To broaden the Web's horizons, I set about giving talks and conducting demonstrations. So that people could see something "out there on the Web" other than the phone book, and to practice what we preached, Robert and I continued to document the project in hypertext on info.cern.ch.

What we had accomplished so far was based on a few key principles learned through hard experience. The idea of universality was key: The basic revelation was that one information space could include them all, giving huge power and consistency. Many of the technical decisions arose from that. The need to encode the name or address of every information object in one URI string was apparent. The need to make all documents in some way "equal" was also essential. The system should not constrain the user; a person should be able to link with equal ease to any document wherever it happened to be stored.

This was a greater revelation than it seemed, because hypertext systems had been limited works. They existed as databases on a floppy disk or a CD-ROM, with internal links between their files. For the Web, the external link is what would allow it to actually become "worldwide." The important design element would be to ensure that when two groups had started to use the