
Contents

- 1. Introduction
 - 1.1 History is Important
 - 1.2 The Major Threads
- 2. CERN
 - 2.1 Atomic Physics: a Very Short Introduction
 - 2.2 Particle Physics
 - 2.3 CERN Accelerators
 - 2.4 CERN Experiments
 - 2.5 CERN Teams
 - 2.6 CERN Computers
 - 2.7 Summary
- 3. Internet
 - 3.1 Introduction
 - 3.2 ARPANET
 - 3.3 Internet
 - 3.4 ISO
 - 3.5 Internet at CERN
- 4. SGML
 - 4.1 Introduction
 - 4.2 Generic Markup
 - 4.3 SGML
 - 4.4 SGML at CERN
- 5. Hypertext
 - 5.1 Introduction
 - 5.2 Vannevar Bush and Doug Engelbart
 - 5.3 Ted Nelson
 - 5.4 Early Hypertext Systems
 - 5.5 Summary
- 6. Tim Berners-Lee
 - 6.1 Introduction
 - 6.2 First Period at CERN
 - 6.3 Second Period at CERN
 - 6.4 Early Years
- 7. World Wide Web
 - 7.1 The Web Arrives
 - 7.2 Web Growth
 - 7.3 E-Commerce
 - 7.4 Countries on the Web

- 8. World Wide Web Consortium
 - 8.1 Introduction
 - 8.2 World Wide Web Consortium
 - 8.3 How W3C Works

Appendices

- A. References
- B. ENQUIRE
- C. Proposal to CERN Management, March 1989 and May 1990

1. Introduction

- [1.1 History is Important](#)
- [1.2 The Major Threads](#)

1.1 History is Important

Understanding how an activity started and developed frequently gives a much greater insight into why things are as they are. In the case of human evolution, it explains why we have certain organs that appear to have no real use in today's world. The **human appendix** is responsible for digestive enzymes that help herbivores process plant material. Thus it may be of use to vegetarians but is not a great deal of help to meat eaters. Understanding evolution explains features such as the appendix which would be otherwise a puzzle. The human coccyx, or **tail bone**," is a group of four or five small vertebrae fused into one bone. The coccyx is a vestige of a tail left over from our monkey-like ancestors.

Plato said that **necessity is the mother of invention**. The world evolves by people having needs that lead to tools or artefacts being developed to satisfy those needs. The needs depend on the tools already available so the arrival of each major tool brings another round of invention and evolution. The wheel, canal, steam engine, train, bicycle, automobile, aeroplane are all major inventions related to people's transportation. There is an equal list of transportation inventions, like the airship, that turn out to be less important but still have value.

When Sir Isaac Newton was being praised for his genius, he responded: **if I have seen further, it is by standing on the shoulders of giants**. The giants were the inventors of previous ages whose contributions had been passed down for him to build on. No one person invented the Web in a vacuum. It depended on the contributions from others who had gone before.

In this Primer we hope to give some insight into the **needs** that led to the invention of the World Wide Web by Tim Berners-Lee in or around 1989. (With all inventions, there is a long gestation period so choosing the precise date for an invention does not make a great deal of sense.) We will also look at the tools that existed to make the invention of the Web possible. We shall also look at the people who developed those tools thus giving Tim Berners-Lee the platform on which to make his own contribution.

With any attempt at understanding history, it depends on the weighting that is placed on specific items. These tend to be subjective so do not expect this brief outline to be definitive. Appendix A gives some references for future study.

1.2 The Major Threads

In this Primer, we will look at the following in some detail:

- **Tim Berners-Lee**: the inventor of the World Wide Web
- **CERN**: the organisation that he worked for that provided the requirements and need
- **Internet**: the infrastructure required early on by the Web
- **SGML**: the basis for document markup on the Web
- **Hypertext**: the underlying linking structure for the Web
- **World Wide Web Consortium**: the organisation whose members develop the specifications used on the Web
- **Early Web Evolution**: how the Web developed in its formative years

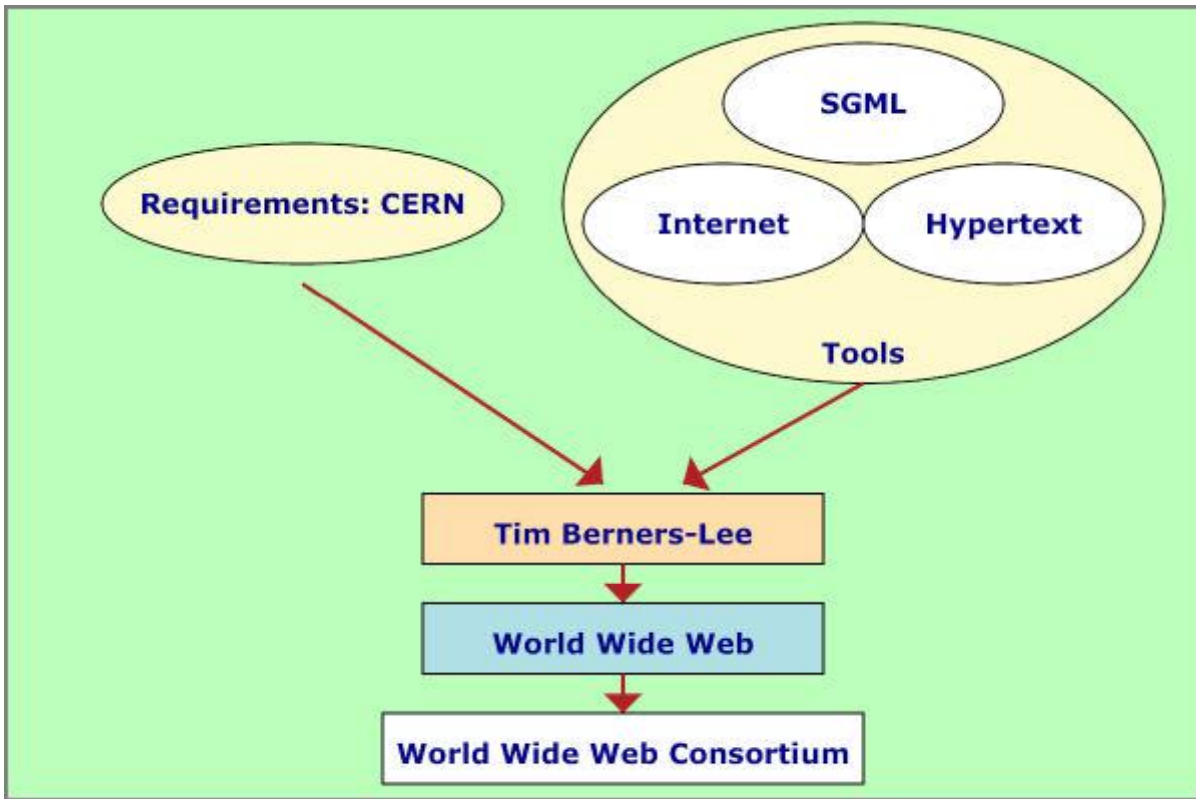


Figure 1.1: Origins of the Web

2. CERN

- [2.1 Atomic Physics: a Very Short Introduction](#)
- [2.2 Particle Physics](#)
- [2.3 CERN Accelerators](#)
- [2.4 CERN Experiments](#)
- [2.5 CERN Teams](#)
- [2.6 CERN Computers](#)
- [2.7 Summary](#)

2.1 Atomic Physics: a Very Short Introduction

In the 19th Century, it was believed that the world was made up of atoms which could be combined together in a great variety of ways to create the compounds that make up the world. These could be as simple as water made up of hydrogen and oxygen atoms through to complex proteins with many hundreds of atoms. The atom was thought to be a spherical blob of more-or-less uniform density. The name atom means **indivisible**.

By 1900, it was realised that the atom was not the most fundamental building block. Rutherford and Geiger discovered that the nucleus was composed of protons and neutrons, both of which weigh almost 2000 times the weight of the electrons that surround the nucleus. The proton is positively charged, the electron negatively charged while the neutron has no electric charge. The existence of the electron was established as early as 1897.

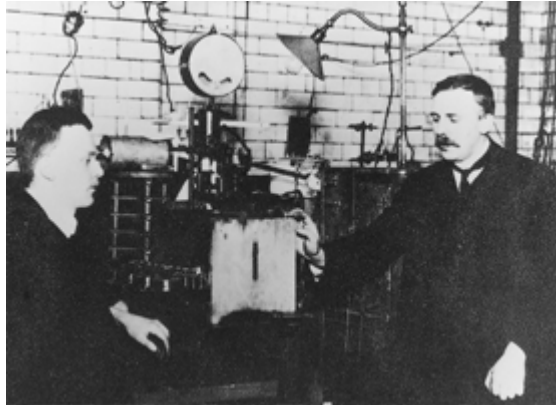


Figure 2.1: Rutherford and Geiger: Discover the Nucleus

One of the greatest achievements of physics in the 20th century was the discovery of quantum mechanics. The exact position of an electron in an atom is not known; the probability of where it might be can be determined. Also, an electron in an atom can only be in one of a well determined set of states. Such discoveries were made without complex apparatus. Even by 1920, most physics laboratories were quite modest with a single person carrying out experiments on a desk top.

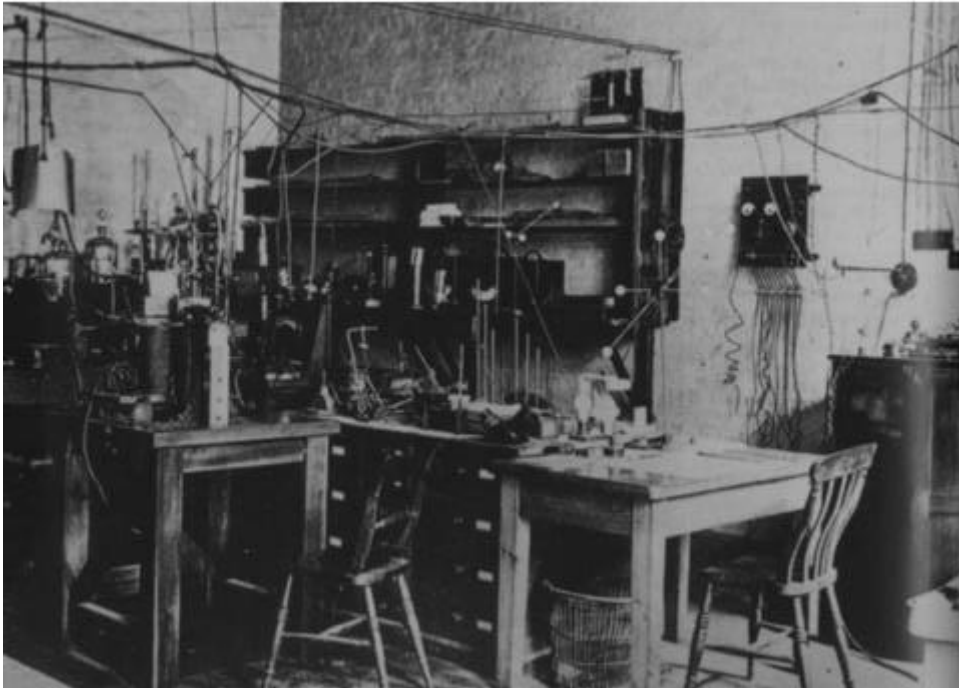


Figure 2.2: Cavendish Laboratory, Cambridge: 1920

2.2 Particle Physics

If a nucleus is made up of positively charged particles, what holds it together? The answer is the **weak** force that operates at distances of the size of the nucleus. One of the challenges has been to unify all the forces of nature, electromagnetism and the weak force into a single theory.

It soon became clear that the nucleus was not made up of indivisible particles called protons and neutrons but these particles were made up of even smaller particles. First one was found and then another. By the 1960s over 100 new particles had been identified. This clearly did not seem very reasonable and the theory was evolving that all these particles were made up of a basic set of particles. Initially it was thought that there were three particles called **quarks** but by the 1990s the theory had become more defined and the belief was that all matter was made up of six basic quarks called **up, down, strange, charm bottom and top quark** and this theory could also unify the forces. The existence of all 6 quarks had been established by 1995.

Given that the nucleus is incredibly small and the particles that the protons and neutrons are made up of are even smaller, it is very difficult to detect them. This is where particle accelerators come in. By getting different particles to hit each other at very high speeds, it is possible to **detect** the results of the collision which often result in the production of other particles. Many of the particles only exist for short period of time before decaying into other particles. In consequence, the detectors have to be very sophisticated to measure the results. Even so, it may take several millions of these collisions before an **event** occurs that is meaningful.

The essence of particle physics is building large high powered accelerators and very precise detectors. Once completed, a specific experiment is run for a very long time, months at least, in the hope that an occurrence of the event that has been theorised takes place.

2.3 CERN Accelerators

After the Second World War, Europe's scientific prestige suffered by the **brain drain** of influential scientists to the USA. In 1949, to redress the balance and restore European science to its former prestige, Louis de Broglie proposed the creation of a European Science Laboratory. By 1952, 11 European governments had agreed to set up a provisional "Conseil Européen pour la Recherche Nucléaire" (CERN) and a site near Geneva was selected for the planned laboratory.

It took until 1957 before CERN's first accelerator, a 600 MeV proton Synchro-Cyclotron began operation. CERN continued to build and run particle physics machines for the next 45 years:

- **1957: Synchro-Cyclotron**
- **1959: PS**, the first major machine, a 28GeV Proton Synchrotron
- **1976: SPS**, a 500 GeV Super Proton Synchrotron
- **1989: LEP**, Large Electron-Positron Collider
- **2005: LHC**, Large Hadron Collider

What can be seen is that the machines take a long while to build and then run for many years before the next machine is built. In 1981, the CERN Council approved the construction of the major new facility, the 27-kilometre Large Electron-Positron collider (LEP) ring, the largest scientific instrument ever constructed. It took until 1989 to build the facility. It was in relation to this new facility that Tim Berners-Lee, working with Robert Cailliau, proposed a distributed information system, based on **hypertext** for CERN groups working on LEP.

Development of CERN's facilities take many years and in 1991 initial work started on a new facility, the Large Hadron Collider (LHC), which will take until 2005 to complete. It will produce data at such a rate that a new infrastructure is needed to process that information. Much of the current hype around GRID computing is concerned with finding ways to process the information from the data hoses connected to the new LHC facility.

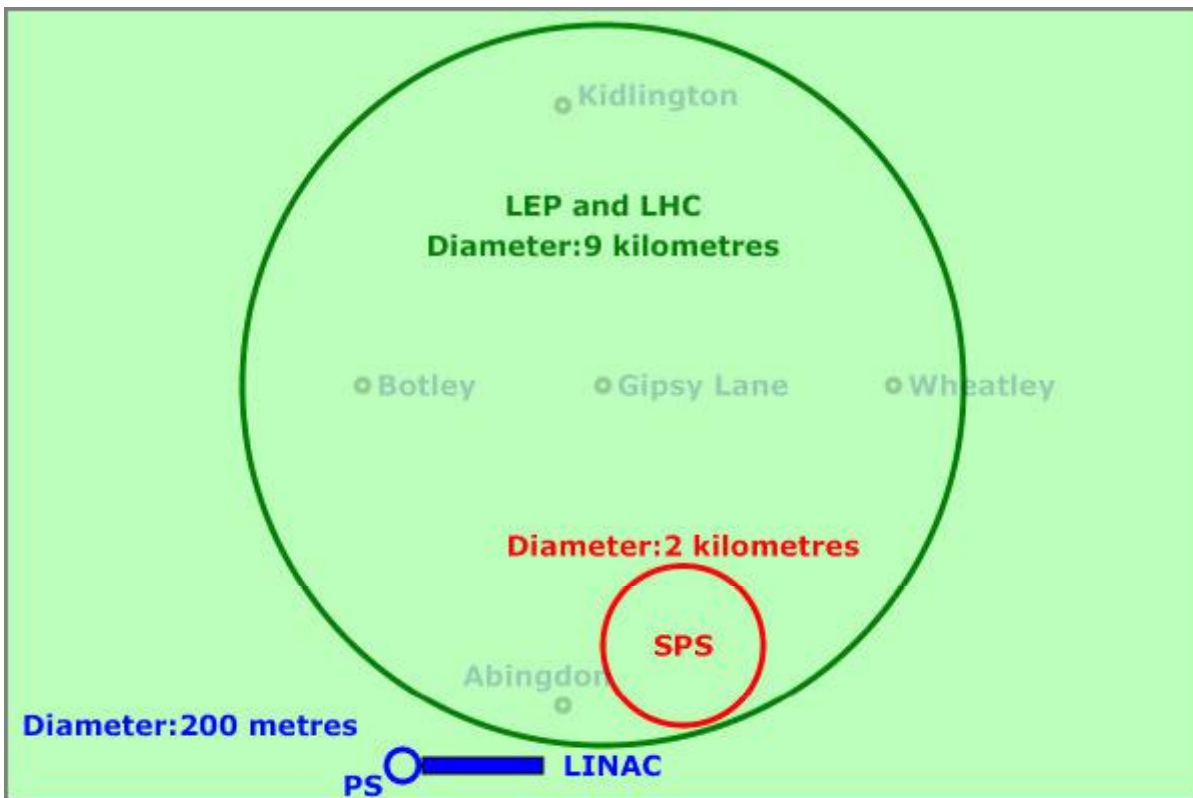


Figure 2.3: CERN Accelerators

Figure 2.3 shows the CERN accelerators. The PS still operates today acting as the particle beam factory feeding the other accelerators with different types of particles. Similarly, the SPS provides beams for LEP. Particles are accelerated in a linear accelerator (LINAC) and fed into the PS which accelerates particles every 2.4 seconds and then injects them into the SPS or LEP. The SPS accelerates protons, collides protons and antiprotons, and acts as an electron/positron injector for LEP. Collisions take place at four symmetric points on the LEP ring. The four LEP experiments are L3, ALEPH, DELPHI, and OPAL

2.4 CERN Experiments

LEP is 100 metres underground and that is where the experiments take place. Bunches of particles rotate in opposite directions and are focused to about the width of a hair and made to collide at the experimental station. Each bunch contains about 100,000,000,000 particles and about 40,000 actually collide each time. So the bunch passes round and round LEP for hours. Each bunch travels around the ring 10,000 times.

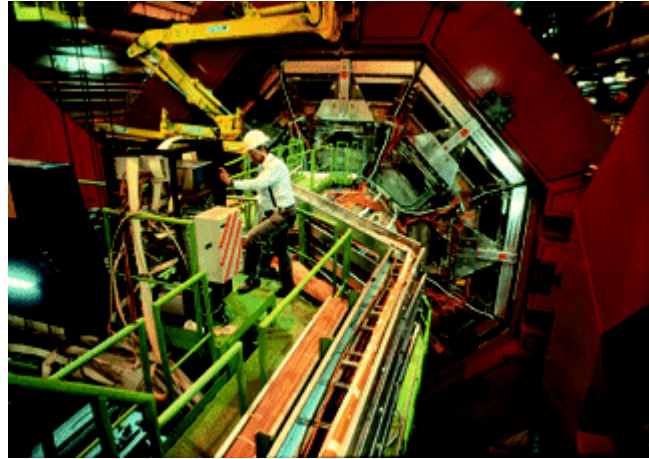
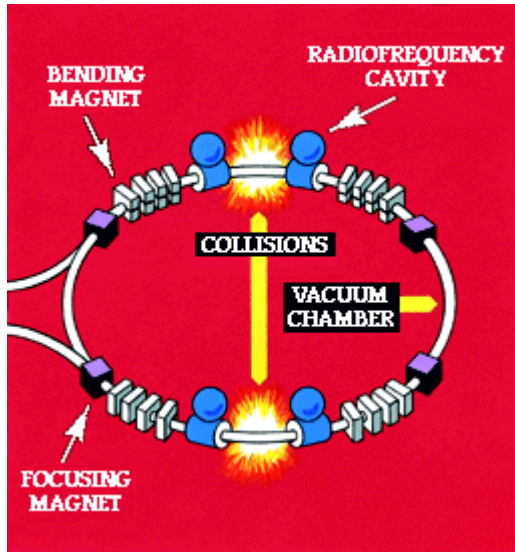


Figure 2.4: Large Electron Positron Collider

Figure 2.5: L3 Detector at CERN

As shown in Figure 2.4, magnets are required to bend the particles around the circle and are also used to focus the beam. The magnets are huge and are specially made for the task. The detectors are as large as three-storey houses (see Figure 2.5, 2.6) as are the magnets .



Figure 2.6: DELPHI Superconducting Solenoid

2.5 CERN Teams

As the machines and detectors get larger, so the teams that build, install and run the equipment get larger. This is no longer experiments done on the laboratory desk top. A scientist devotes a significant portion of his working life to a single experiment. Even when the experiment is complete, several years may still elapse before all the data has been analysed. The largest experiment on the SPS had 224 people working on it from 23 different institutions. This compares with the LEP experiments that have between 300 and 700 people working on them from 30 to 40 institutions. Figure 2.7 shows the distribution of one of the CERN Teams of 300 physicists situated in 34 institutions worldwide.

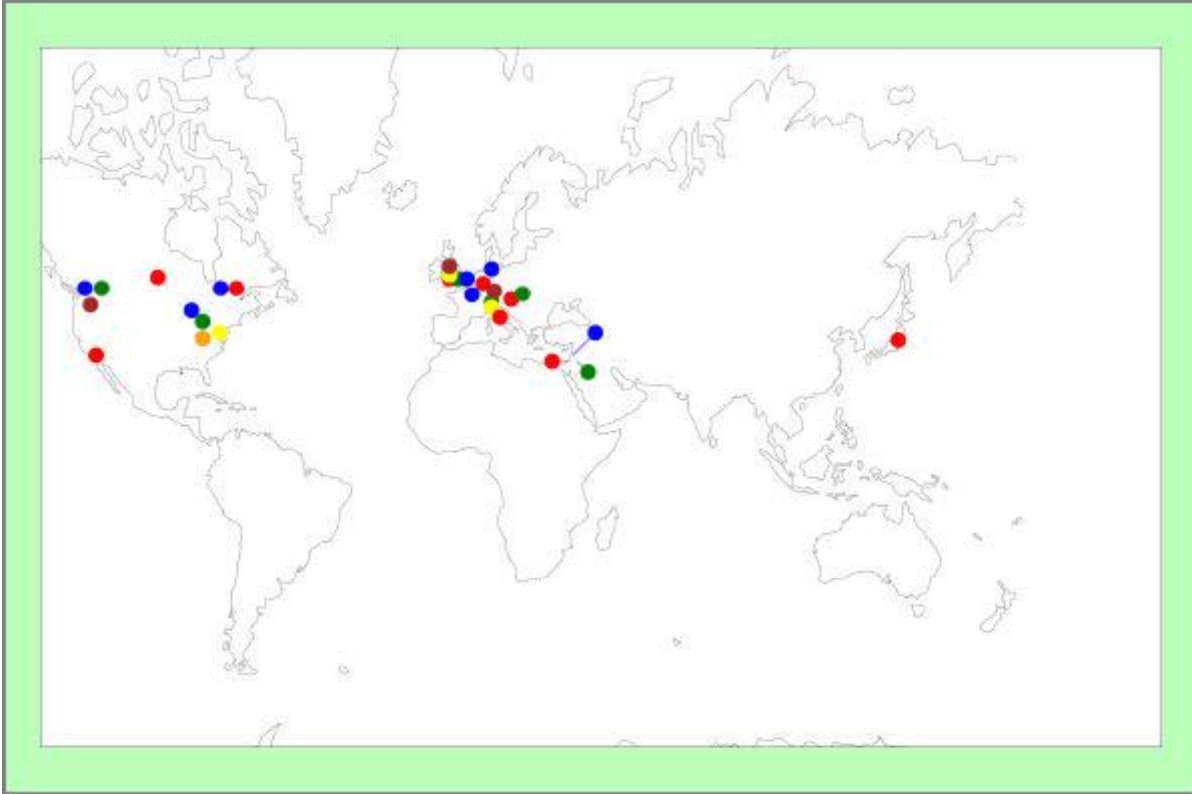


Figure 2.7: Geographic Distribution of a CERN Team

2.6 CERN Computers

The first accelerators installed at CERN were controlled by purpose-built hardware with knobs and dials tweaked by operators. By 1966, PS started getting computers added to control certain functions or to capture data from the experiments. Prior to that, results were often photographic film scanned off line by large teams of human scanners at the institutes from which the CERN team came. The SPS was the first accelerator with a fully computerised control system. Around, 1980 the PS system was computerised to allow it to act as a flexible particle factory for the SPS. Tim Berners-Lee, in his first period at CERN worked on applications for the PS.

The computerisation of LEP far exceeded what had been done on earlier accelerators. The need to distill the information coming from the detectors arose due to the sheer volume being generated. OPAL, the smallest LEP experiment, stood for Omni-Purpose Apparatus at LEP and consisted of 15 sub-detectors each under the control of a computer which was connected to a hub computer that assigned tasks. The sub-detectors were produced by different parts of the team and so there was likely to be different computers running different operating systems. Thus the interest in remote procedure calls as a standard way of controlling the sub-detectors from the hub. This was the area in which Tim Berners-Lee worked on his second period at CERN.

2.7 Summary

CERN in the 1980s was an international organisation with large teams developing complex apparatus spread across several continents. Ensuring that everybody in the team had up-to-date information about what was going on was of utmost importance. The distributed nature of the development and the sheer size and complexity meant that hundreds of computers networked together were used to control the accelerators, the experiments and collect the data. These were the requirements that inspired the invention of the World Wide Web.

3. Internet

- [3.1 Introduction](#)
- [3.2 ARPANET](#)
- [3.3 Internet](#)
- [3.4 ISO](#)
- [3.5 Internet at CERN](#)

3.1 Introduction

In the early days of computing, programmers produced their programs on punched cards or paper tape and gave them to an operator to run on the computer. If the computer was remote, the programmer had to physically visit the computer site or use a courier to send his program to be run.

In the early 1960s, the large computers started to get small computers to act as **remote job entry (RJE)** facilities. Initially the RJE facilities spooled the jobs to be run on to magnetic tape to be transported to the main computer. Similarly, the results would be put on to a magnetic tape and it was returned for local printing at the RJE station. By the end of the 1960s, the RJE stations were connected by telephone lines to the main computer thus removing the need to transport magnetic tapes backwards and forwards. Even so, for large volumes of data, carrying handbaggage full of magnetic tapes was still the fastest mechanism for moving data for another 20 years. On the CERN site, bicycles were used for the same purpose!

Around 1965, the concept of running several programs on the computer at the same time had evolved. Initially, this was just spooling the input and output, later it was running a second program when another was waiting input and finally programs were **time-shared** on the computer executing for a short period before being swapped for another to have its go. Once the computer could be time-shared it was feasible to provide interactive use. Systems were developed which initially allowed users to edit their programs online while batch programs were being executed in the background. Later, true interactive use of the mainframe computers was allowed. But these systems were expensive and did not perform well with a large number of users being scheduled together.

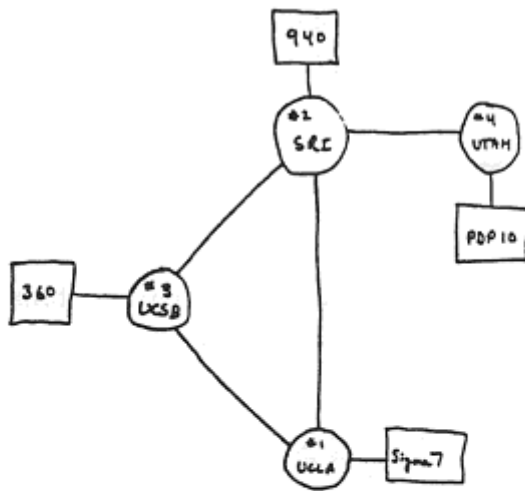
By 1975, it became feasible to purchase inexpensive systems from DEC, Prime, Interdata, GEC and others that provided good interactive facilities for perhaps 10 people at a time. There was still the need to interact with the large mainframe systems for running large simulations etc. In consequence, a set of proprietary systems were developed (DECNET, PRIMENET, etc). These allowed a user on one machine to access and run programs on another. Some of the standard office functions such as email, diary handling, document repositories appeared but all heavily proprietary with little chance of interoperability. They all worked on the packet-switching approach of not sending whole messages in one go but breaking up a message into packets which were interleaved with the packets from other users.

It was at this stage that events in the USA and Europe diverged. Europe spent a great deal of time and effort in attempting to provide an international standard for packet switching via ISO while the ARPANET was developed in the USA. ISO took from 1968 to 1976 to formulate the X.25 OSI standard while between 1967 and 1969 ARPANET became operational.

3.2 ARPANET

ARPA, the Advanced Projects Research Agency, had by 1962 become the organisation responsible for funding long-term research in the USA. It had an Information Processing Techniques Office, headed by Lick Licklider, that had funded Project MAC at MIT, one of the first examples of time-sharing. Licklider was followed by Ivan Sutherland and later Bob Taylor, who hired Larry Roberts as Project Manager to oversee networking. There were only a limited number of large, powerful research computers in the country, and many research investigators who should have access to them were geographically separated from them. The networking project was designed to solve that problem.

The initial problem was to connect the main large computers most of which ran different operating systems. The decision was made to define an Interface Message Processor (IMP) that would sit in front of each large computer. The IMP would be responsible for packet switching services between the computers. This standardised the network which the hosts connected to. Only the connection of the hosts to the network would be different for each type of computer. ARPA picked 19 possible participants in what was now known as the **ARPA Network** later to become ARPANET.



THE ARPA NETWORK

DEC 1969

4 NODES



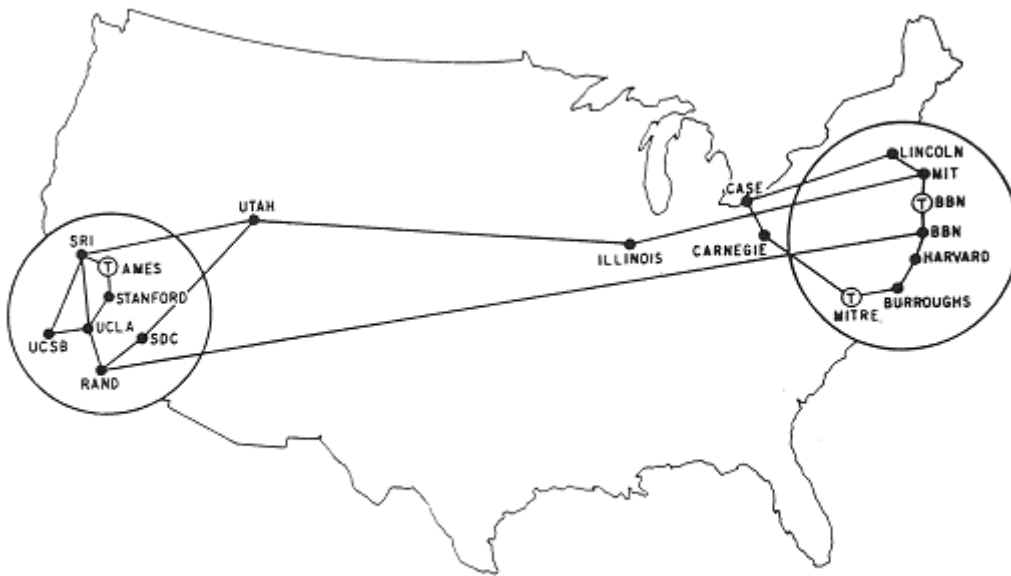
FIGURE 6.2 Drawing of 4 Node Network
(Courtesy of Alex McKenzie)

Figure 3.1: ARPANET: 1969

BBN won the contract to develop the IMP-to-IMP subnetwork. The protocols to be used was left **for host sites to work out among themselves**. The first sites of the ARPANET were picked on the basis of networking competence and either providing network support services or unique resources. The first four sites were:

- **UCLA:** Network Measurement Center
- **SRI:** Network Information Center
- **UCSB:** Culler-Fried interactive mathematics
- **Utah:** 3D graphics (hidden line removal)

Meetings took place in 1968 to define the protocols. Ideas were produced as **Request for Comments (RFCs)** and this continues even today. The first IMP was delivered on 30 August, 1969 to UCLA and the second to SRI in October. By 1971, the ARPANET had grown to 19 nodes running a protocol called NCP (Network Control Protocol).



MAP 4 September 1971

Figure 3.2: ARPANET: 1971

In 1970, local area networks (LANs) started to appear, in particular **ethernet** at Xerox PARC. This led to a need for a protocol that would work across networks. Wireless networks were appearing that could not guarantee that packets would arrive in the correct order due to the need for retransmission of some packets. This led to the work on a host-to-host **Transfer Control Protocol** for internetworking. The first specification of TCP was published in December 1974 and concurrent implementations were done at Stanford, BBN, and Peter Kirstein's group at University College London. TCP eventually became **TCP/IP**, two protocols instead of one where IP stands for **Internet Protocol**. IP was mainly responsible for the routing of packets over the network while TCP checked they had arrived and reassembled them in the correct order.

In networking terms, IP is a connection-less protocol capable of moving information between LANs and WANs where each computer had a **unique address**. TCP is a connection-oriented protocol at the transport level, in OSI terms.

3.3 Internet

The birth of the Internet is usually given as July 1977 when there was a large scale demonstration of internetworking using the ARPANET, packet radio networks and satellite transmission. A message was sent from a van on the San Francisco highway by radio to ARPANET. From there, it went by satellite to Norway, by a land line to UCL, by satellite back to the USA and by ARPANET to Los Angeles. A round trip of 94000 miles across several networks and technologies to deliver a message 800 miles away!

ARPANET GEOGRAPHIC MAP, OCTOBER 1980

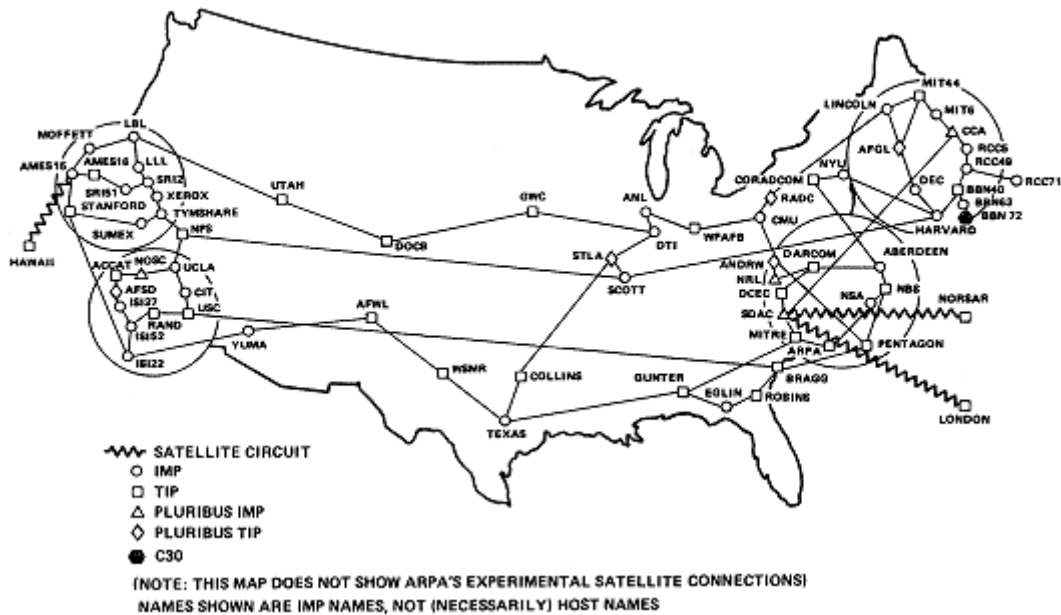


Figure 3.3: ARPANET: 1980

In the original version of TCP, the internet addressing consisted of 32 bits, 8 of which defined the network. Thus the maximum number of networks allowed was 256. While this might have been sensible in the days of mainframes, the arrival of smaller systems and later workstation and PCs meant that this was not going to be sufficient. By the end of 1982, the experimental use of TCP/IP was complete and TCP/IP was launched as **THE** Internet protocol in January 1983. The same year saw the move to the hierarchical Domain Name System (DNS) used today. By 1985 there were 2000 hosts attached to the Internet and by 1990 there were 2000 networks which had grown to 94,000 networks by 1996.

3.4 ISO

While the USA was going with TCP/IP, most of Europe was trying to base their future internetworking on the set of protocols being defined by ISO called Open System Interconnect (OSI). By 1983, the 7-layer reference model for OSI had been defined:

- **Layer 7 Application:** Communication partners identified, quality of service, user authentication
- **Layer 6 Presentation:** Converts incoming and outgoing data from one presentation format to another, character codes etc
- **Layer 5 Session:** Sets up, coordinates, and terminates exchanges between the applications at each end.
- **Layer 4 Transport:** Manages the end-to-end control, breaks into packets, have all packets arrived. Ensures complete data transfer.
- **Layer 3 Network:** Handles the routing and forwarding of the data
- **Layer 2 Data link:** Error control and synchronization for the physical level. Transmission protocol knowledge and management.
- **Layer 1 Physical:** Conveys the bit stream through the network at the electrical and mechanical level. Provides the hardware means of sending and receiving data on a carrier.

Figure 3.4 shows the 7 layers of the ISO model and how interaction would occur with a non-OSI network. The fact that TCP/IP does not have the same conceptual basis does not make it easy.

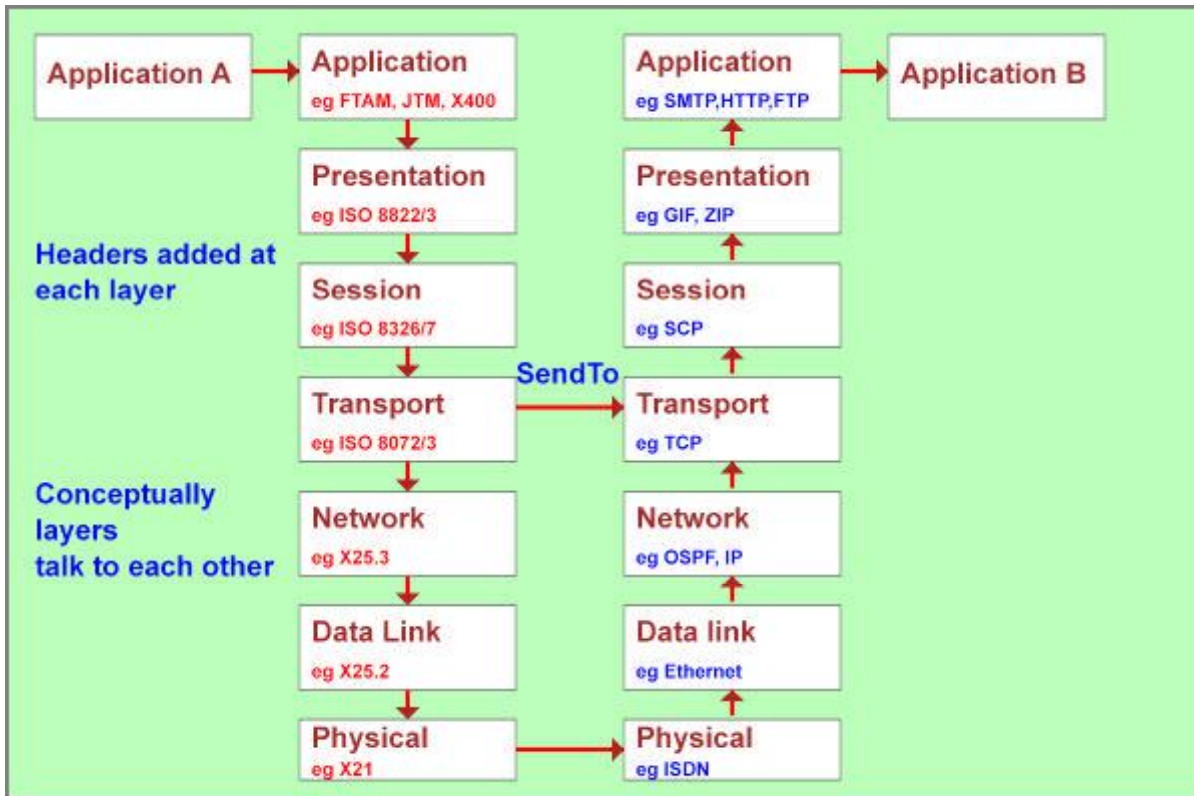


Figure 3.4: ISO 7-Layer Model

It is probably not worth wasting a lot of time on OSI. The protocols were probably better defined than TCP/IP but the slowness of the ISO process and the reluctance of USA-based companies to implement OSI without being paid made the uptake slow and patchy. The UK set up the Joint Network Team (JNT) to push OSI protocols in the academic community. The JNT introduced a set of Coloured Book Protocols as early implementations of the OSI protocols as they were developed. On the Local Area Network side, a lot of effort was put into backing the Cambridge Ring as an alternative to ethernet. A pilot TCP/IP service was not implemented until 1991.

3.5 Internet at CERN

In the 1970s, networking at CERN was a mixture of home grown solutions and proprietary ones. CERNET was a packet switching fast file transfer service between mainframes and minicomputers using locally defined protocols. IP was first used at CERN in the period 1981-83 as the basis for a satellite communication system between the Rutherford Appleton Laboratory (RAL), CERN and Pisa. Ethernet appeared at CERN in 1983. CERN was caught up in the move to ISO standards in Europe and spent a few years following the party line. By 1984, a decision was made to set up a pilot implementation of TCP/IP. Ben Segal was the TCP/IP Coordinator and by the end of 1985 it had been agreed that LEP communication would be based on TCP/IP. In 1988, TCP/IP became a supported infrastructure at CERN and CERN was connected to the Internet in January 1989. By 1990, CERN was the largest Internet site in Europe.

A key result was that by 1989, Tim Berners-Lee could create the World Wide Web based on **distributed computing** and the Internet.

4. SGML

- [4.1 Introduction](#)
- [4.2 Generic Markup](#)
- [4.3 SGML](#)
- [4.4 SGML at CERN](#)

4.1 Introduction

A fundamental part of the Web is the presentation of textual information. Document markup is the process of inserting commands in a document describing the structure of a document or the format in which it is to appear when printed. Document markup is a communication form that has existed for many years. Early on, markup was done by a copy editor adding commands to a manuscript for a human typesetter to follow. A standard set of symbols was eventually used by copy editors to do this communication.

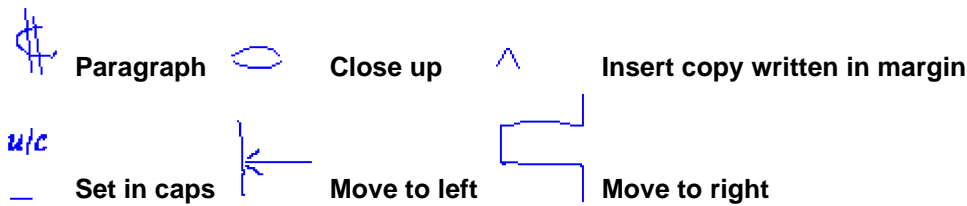


Figure 4.1: Copy Editor Markup

When computers started to be used for textual presentation, text formatting languages were defined and used by the human typesetter. Eventually the author became proficient at adding the text formatting instructions themselves. As with networking, proprietary systems appeared each with their own markup language. The advent of printing allowed a consistent page layout for a book. Prior to that, hand written books would vary on number of words per page dependent on the scribe. In consequence, meta information such as Contents pages and an Index were late in arriving. By 1450, single letter sorting of Indexes was all that the reader had. So an added benefit of computerising the markup of a document was that such metadata became easier to produce as well.

Two events that really sparked the use of computers for typesetting was the arrival of the IBM Golfball typewriter and the appearance of timesharing systems. The first gave typewriter quality output and the second allowed interactive editing and markup. In 1964, the CTSS Timesharing System at MIT had a system called RUNOFF which allowed the user to add typesetting commands to the document:

```
.ctr Primitive Markup Title
.sp 2
.nj
Some code not to be formatted
.bp
This is on a new page
```

As with networking, many proprietary systems arrived. Each manufacturer had their own language for adding markup and it was very difficult to switch between them. A major change with the arrival of Unix as a common operating system and with it a common markup facility initially called **roff** and later **nroff** for **new** roff. So the period 1967 to 1975 saw a measure of a standard and this was replaced by a more widely used standard, **troff** in 1975. It had two important features over the earlier systems. The first was device independence and the second was a macro facility. The user was now able to define his own markup in a more meaningful way leaving the precise typesetting to be achieved being part of the macro. Associated with troff was also a set of preprocessors that made the production of tables (tb), pictures (pic), equations (eqn) and graphs (grap) easier to do.

4.2 Generic Markup

The main problem with the markup defined in the previous section is that the markup provides both information about the **content** of the document and how it is to be **presented**. The RUNOFF commands are mainly concerned with presentation while the macro facilities within troff give the ability to indicate, say, that the **text** is a Chapter Heading and how the heading should be presented.

The advent of generic coding was a presentation made by William Tunnicliffe of the Graphic Communications Association (GCA) to a meeting of the Canadian Government Printing Office in September 1967 entitled **The separation of information content of documents from their format**.

The image shows a page from a Yellow Pages directory. On the left side, there is a list of publishers with their names, addresses, and phone numbers. The list includes companies like 'Oxford Publishing Ventures Ltd.', 'Oxford University Guild Publications', and 'Pulley & tackle mfrs'. On the right side, there are several advertisements. One is for 'apre pumps' with contact information 'TEL: 01494 837268'. Another is for 'BROADWAY PUMPS' with the tagline 'SUBMERSIBLE PUMP SPECIALISTS'. A third is for 'Dennis Roberts Ltd' with the headline 'Tying the knot? Wedding Guide see Wedding Services'. The ads are for various types of pumps and equipment, including agricultural and industrial pumps.

Figure 4.2: Yellow pages

Bill Tunnicliffe had his own printing company and one of the jobs they did regularly was typeset the local Yellow Pages (Figure 4.2). This is a publication dense in information that partly changes each edition but much information stays the same. Having the presentation markup interspersed in the document itself means that each edition has to be typeset from scratch. Also, the typesetting does not have to be that precise. Figure 4.2 shows a gap on the right side that the typesetter has had difficulty filling and so throws in a local advert to fill the space.

GCA established a generic coding project which proposed GenCode. GenCode defined a generalised markup approach based on a document's hierarchy and attempted to standardise a set of common elements.

In 1969, Charles Goldfarb of IBM with Edward Mosher and Raymond Lorie started work on the Generalized Markup Language. It allowed you to define your own markup terms and a validating parser. For the first time the **tagging** was hierarchical with both start and end tags enclosing the marked up text. You could say what markup was allowed where and markup tags could be omitted if their position was obvious. By 1973, IBM had an implementation of GML as part of its Advanced text Management System. An example of GML is:

```
:book.
:body.
:h1.Introduction
:p.GML supported hierarchical containers, such as
:ol.
:li.Ordered lists (like this one),
:li.Unordered lists, and
:li.Definition lists
:eol.
:p.as well as simple structures.
:p.Markup minimization allowed the end-tags to be omitted for the "h1" and
"p" elements.
```

GML used the symbols colon and fullstop as tag delimiters but used **h1** to define major headings, **p** for paragraph, **ol** for an order list and **li** for a list element. For those with a knowledge of HTML, these should be quite familiar. However, end tags did not use the **:ol.** notation for the end tag but had **:eol.** GML had demonstrated that it was possible to markup information in a generic way so that it could be used by more than one application.

4.3 SGML

In 1978, Goldfarb led a project for ANSI to produce an American standard text description language based on GML. This became SGML (Standard Generalized Markup Language) and was moved to ISO to become an international standard. The first working draft of the SGML standard was published in 1980 and by 1983 the sixth working draft was adopted by the US Internal Revenue Service (IRS) and the US Department of Defense. It finally became an ISO standard in 1986 with the standard, of course, produced in SGML using a system developed by Anders Berglund at CERN. One of the earliest books was **Practical SGML by Eric van Herwijnen** of CERN.

SGML provided the mechanisms to define a markup language for a particular purpose. It did not constrain the tags that could be used to markup the text nor did it constrain what symbols would be used for the opening start-tag, opening end-tag, closing start-tag and closing end-tag delimiters. A simple memo could be marked up as:

```
:memo)
:to)Fred Bloggs*to]
:from)Joe Smith*from]
:subject)Lunch*subject]
:body)
:p)Do you want to have lunch?*p]
:p)I am free at 12.00*p]
*body]
:signature)Fred*signature]
*memo]
```

The symbol **:** has been used as the open start-tag, **)** as the closing start-tag, ***** as the opening end-tag and **]** as the closing end-tag. This is called the **concrete syntax** that is being used by SGML. In the standard, a particular concrete syntax was used as follows:

```
<memo>
<to>Fred Bloggs</to>
<from>Joe Smith</from>
<subject>Lunch</subject>
<body>
<p>Do you want to have lunch?</p>
<p>I am free at 12.00</p>
</body>
</signature>Fred</signature>
</memo>
```

As well as defining the syntax for the markup and the concrete syntax, SGML also defined a **Document Type Definition** which accompanies the document and defines the structure of tags that are allowed. In the example above, the DTD would say that the outer element is **memo**. It must be followed by **to**, **from**, **subject** and **body** elements in that order and that the **body** element can consist of a set of **p** elements etc. The DTD could also say which end tags could be omitted. ISO also defined a separate standard called **DSSSL**, the Document Style Semantics and Specification Language which described the presentation formatting required by the document's author.

4.4 SGML at CERN

SGML was widely used at CERN for documentation starting as early as 1984. A documentation system developed at RAL on an IBM mainframe was rewritten as CERNDOC with SGML as its basis.

The SGML application used at CERN was derived from the American Association of Publishers (AAP)'s DTD for marking up books. CERN SGML documents were defined in the CERN SGML GUID language. Here is an example of the markup from 1990 that predates the Web and has an SGML ending and is written by Tim Berners-Lee:

```
<BODY>
<H1>Introduction
This manual describes how to build a distributed system using the
Remote Procedure Call system developed in the Online Group
of the DD Division of CERN, the European Particle Physics Laboratory.
<h2> The system
The remote procedure call product consists of two essential parts:
an RPC compiler which is used during development of an application,
and the RPC run time system, which is part of the run time code.

Target systems supported are
<ul>
<li>VAX/VMS,
<li>Unix (Berkley 4.3 or Ultrix or equivalent)
<li>stand-alone M680x0 (MoniCa) systems (Valet-Plus, etc)
<li>stand-alone M6809 systems
<li>M680x0 systems running RMS68K
<li>M680x0 systems running OS9
<li>The IBM-PC running TurboPascal or Turbo-C
<li>The Macintosh running TurboPascal or MPW
</ul>
```

This was written in 1986 and last modified in April 1990.

5. Hypertext

- [5.1 Introduction](#)
- [5.2 Vannevar Bush and Doug Engelbart](#)
- [5.3 Ted Nelson](#)
- [5.4 Other Hypertext Systems](#)
- [5.5 Summary](#)

5.1 Introduction

Hypertext is about linking information together. There is nothing very new about hypertext. The Alexandria Library of 300BC used annotations and references between works. In AD 79 Pliny the Elder produced **The Natural history** in 37 volumes and Book One was a Table of Contents. While books were still being written by hand, it was difficult to afford the additional cost of adding links. By 1450, the Index had started to appear. Dictionaries were in general use by the 16th Century. The volume of information in the 20th Century increased their importance and the arrival of the computer made more exciting uses possible. The term **hypertext** is defined as non-sequentially linked pieces of information.

5.2 Vannevar Bush and Doug Engelbart

In 1927, Vannevar Bush designed an analog computer that could solve simple equations. A bigger version called the **Differential Analyser** was developed in 1930 and by 1935 the mechanical version was replaced by an electro-mechanical version. It was a modest machine weighing 100 tons and contained 2000 valves, 150 motors and 200 miles of wire. During the 1939-45 war, the Differential Analyser was used by the team developing the atomic bomb. One was installed at Manchester University.

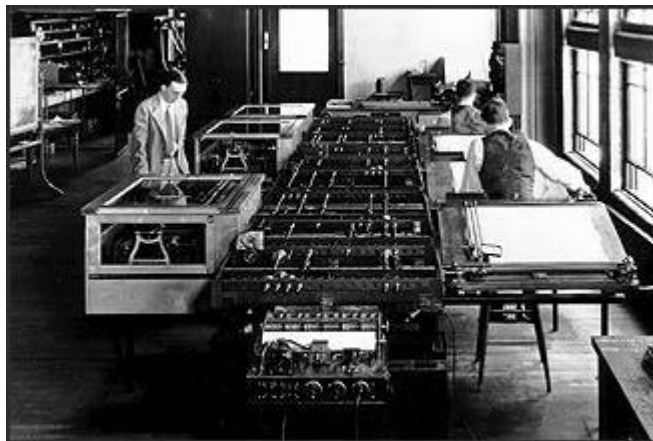


Figure 5.1: Vannevar Bush's Differential Analyser

Bush then developed the Rapid Selector, a microfilm storage and information retrieval device. By 1945, he was Director of the Office of Scientific Research and Development, in the USA coordinating the activities of 6000 leading American scientists in the application of science to warfare. In July 1945, he wrote a paper **As We May Think** for the Atlantic Monthly which said now that the war was over we should look forward to the future. As the human mind worked by association, he suggested that we should build a **memex**, a device in which an individual stores all his books, records, and communications, and which is mechanized so that it may be consulted with exceeding speed and flexibility. It is an enlarged intimate supplement to his memory. The concept of **hypertext** had been invented. Bush went on to say:

Wholly new forms of encyclopedias will appear, ready-made with a mesh of associative trails running through them, ready to be dropped into the memex and there amplified. The lawyer has at his touch the associated opinions and decisions of his whole experience, and of the experience of friends and authorities. The patent attorney has on call the millions of issued patents, with familiar trails to every point of his client's interest. The physician, puzzled by its patient's reactions, strikes the trail established in studying an earlier similar case, and runs rapidly through analogous case histories, with side references to the classics for the pertinent anatomy and histology. The chemist, struggling with the synthesis of an organic compound, has all the chemical literature before him in his laboratory, with trails following the analogies of compounds, and side trails to their physical and chemical behavior.

Bush was particularly worried about the explosion in scientific literature occurring at that time. Just as tools such as the microscope had enhanced human perception so there was a need for a tool to help the human organise and understand the volumes of information that were appearing.

In 1945, Doug Englebart was in the navy and left San Francisco bound for the Philippines. The war ended as he sailed. He came across the article by Bush while in the Philippines. He was influenced by reading Bush's paper and started his own long term research aimed at augmenting human behaviour. He started his own research lab, the Augmentation Research Center and developed the NLS (oNLine System) aimed at structured information that could be easily and immediately manipulated. This system was used to store all research papers, memos, and reports in a shared workspace that could be cross-referenced with each other. In 1968, a famous demonstration of NLS showed the world the **mouse**, cooperative working, the use of ARPANET, and linked information.

In 1965, Ted Nelson coined the word **hypertext** for linked information. He used the definition:

A body of written or pictorial material interconnected in a complex way that it could not be conveniently represented on paper. It may contain summaries or maps of its contents and their interrelations; it may contain annotations, additions and footnotes from scholars who have examined it.

Modern hypertext (or hypermedia or multimedia, the terms are not that precise) systems are designed to allow people to create, annotate, link together, and share information from a variety of media such as text, graphics, audio, video, animation, and programs. The key point is that hypertext systems provide a non-sequential method of accessing information unlike traditional information systems which are primarily sequential in nature. That is not totally true, of course, as the whole of computer graphics has worked in this manner for the last 45 years so the emphasis has to be on the problems with linking text if it is a **hypertext** system.

Hypertext systems tend to think of information as nodes (concepts) connected by links (relationships). They will have a way of editing and managing both the nodes and the links. They are usually complete systems which hold all the nodes and links in a robust database.

5.3 Ted Nelson and Xanadu

Ted Nelson was influenced by Bush's work on the Memex and for over 40 years he has been working on his Xanadu system with the vision of a document universe where **everything should be available to everyone. Any user should be able to follow origins and links of material across boundaries of documents, servers, networks, and individual implementations. There should be a unified environment available to everyone providing access to this whole space.**

By 1972, Xanadu was running with both Fortran and Algol implementations. It had a novel set of data structures and associated algorithms called the **enfilade** (dictionary meaning: to spin out; to put thread in a needle, to string beads by putting them on a thread). Information was not stored in files but **zipper lists**. The system was completely redesigned in 1981 and the Xanadu Operating Company (XOC) was launched in 1983. XOC was acquired by Autodesk in 1988 with the aim of turning it into a commercial product. After spending \$5M, Autodesk dropped the project in 1992.



Figure 5.2: Ted Nelson

Xanadu always had a strong interest in Copyright and how authors would get payments in a hypertext world. Ted Nelson continues to work on Web-based system concerned with Copyright and micropayments.

Ted Nelson is currently at Keio University in Japan working on a new system called ZigZag. This is a multi-dimensional environment for storing information where the user can focus on any 3 dimensions at a time with two being presented in the X and Y dimensions and the third hidden. You might have a diary where moving down the page changes the timeline while moving left and right changes the granularity of the information. On Ted Nelson's Home Page is a link to **The New Xanadu for the Web**. Traversing it in July 2001 gives you: **404: This page will be available sometime in July.**

5.4 Early Hypertext Systems

Some of the early hypertext systems were:

- **HES:** In 1967, Andy Van Dam at Brown University developed a system with Ted Nelson called HES for Hypertext Editing System. It was used by Houston to produce documentation for the Apollo Space Programme. Arbitrary length pieces of text could be linked together (unidirectional links). It allowed boiler plate text to be edited and the results appear in all the documents that used that text.
- **FRESS:** Andy built a new system soon after called FRESS (File Retrieval and Editing System) which was based on NLS and ran on an IBM mainframe. It separated the internal formats from the I/O by having virtual input/output devices. It had bidirectional links so that it was possible to know what people were linking to your information. Links could be typed to indicate their purpose and protection was implemented down to the byte level. It was one of the first systems to have an **undo** operation.
- **ZOG and KMS:** In the early 1980s, Carnegie Mellon University developed a document management system called ZOG which ran on a PERQ workstation, the first commercial workstation with a bit map display. It was used on the Carl Vincent battleship. By 1983 this had developed into KMS (Knowledge Management System) which ran on UNIX workstations. It had a simple data structure called a **frame**. Linking was made between frames and the concept of going back to earlier frames was introduced.
- **Intermedia:** In 1985, Norman Meyrowitz at Brown developed a new system called Intermedia. It kept the links in a separate database so that different users could have different views on the same material. Brown eventually spun off a company, Electronic Book Technologies which produced a commercial product, DynatText.
- **GUIDE:** In the UK, Peter Brown's GUIDE system was developed around 1985 and became one of the first commercial products aimed at documentation marketed by Office Workstations Limited (OWL). It supported different kinds of links. **Inline replacement** links provided a hierarchical structure like a book. Thus a section heading in an index could be replaced by the contents of the section. **Pop-up** links opened a separate window and were much like annotations. The third type were **reference** links that took you to another location.
- **Hypercard:** Probably the most influential hypertext system was Bill Atkinson's HyperCard for the Mac computer. Introduced in 1987, HyperCard organises information into **stacks** of **cards** through which users can navigate and search for the information they need. Clicking on a button directs the user to a related card. It had an easy to use development system for new stacks and is widely used even today. Apple has discontinued the product but it refuses to die. Robert Cailliau at CERN was an early HyperCard user.

By 1987, there was sufficient interest in the topic to have the First Hypertext Conference at the University of North Carolina with 300 people taking part (although others were turned away). Andy Van Dam gave the keynote talk and listed a number of current issues including the need for standards, accessibility, version control, and handling real world size hypertext systems. All of the current systems were toy systems or worked in a very specific domain. There was a need for a system that would scale to handle very large information spaces.

Two current European Hypertext Systems are Wendy Hall's Microcosm and the Hyper-G system from the University of Graz.

Hypertext systems have many of the problems associated with multimedia systems. Writing a good system is difficult for the author. Users get disorientated with the **lost in hyperspace** syndrome. Consequently many systems spend a lot of effort keeping trails where the user has been, providing overviews of the system, giving guided tours, providing fixed trails through the information etc.

5.5 Summary

When Tim Berners-Lee invented the Web in 1989, there had been a long interest in hypertext systems. Some commercial products had started to appear but were aimed at niche markets. In particular, learning Systems, museum exhibits, and interactive kiosks had all been successfully developed and there was an Annual Hypertext Conference. No large scale hypertext systems existed.

6. Tim Berners-Lee

- [6.1 Introduction](#)
- [6.2 First Period at CERN](#)
- [6.3 Second Period at CERN](#)
- [6.4 Early Years](#)

6.1 Introduction

In 1948, Manchester University invented the stored program computer, the **Baby**. This was followed by the Manchester Mark 1 in 1949 and a commercial version of that was produced as the Ferranti Mark 1. Mary Lee and Conway Berners-Lee both worked on developing the Ferranti Mark 1 and in 1955 Tim Berners-Lee was born. Figure 6.1 shows Conway in 1999 (second from left) with some others who worked on the early Ferranti machines. The first Ferranti machine went to Manchester University in 1951 with Mary setting it up, probably making her the first commercial programmer. Nine systems were sold.



Figure 6.1: Conway Berners-Lee and the Ferranti Mark 1

Tim Berners-Lee was awarded a First Class Physics Degree at Queen's College, Oxford in 1976. In those days, companies visited universities looking for recruits in the so-called milk round. Plessey Telecommunications Ltd based at Poole won on the basis of sun and countryside. He spent two years there working on distributed transaction systems, message relays, and bar code technology. In 1978 Tim joined D.G Nash Ltd (Ferndown, Dorset, UK) which was run by two of his friends Dennis Nash and John Poole. He wrote among other things typesetting software for intelligent printers, and a multitasking operating system. He added a microprocessor to a dot matrix printer to do graphics. He then worked as an independent consultant with John Poole and Kevin Rogers, his best man when he got married to Jane Northcote. After his first period as a contract programmer at CERN, they started up a company, Image Computer Systems which marketed the intelligent dot matrix printer. Poole invited Tim back as a Director of the new company and he worked on graphics and communications software. Tim writes [1]: *We rewrote all the motor controls to optimise the movement of the print head so it was fast. It would print Arabic, draw three-dimensional pictures, and give the effect of preprinted stationery, while using less expensive paper. We wrote our own markup language in which documents were prepared, and the printer could also handle input codes of much more expensive typesetting machines. We could change not only fonts but almost any aspect of the printer's behaviour.* The company still exists and has been at the forefront of barcode, label and retail sign printing ever since 1979.

6.2 First Period at CERN: ENQUIRE

Tim's first period at CERN started on 23 June 1980. Benney Electronics of Southampton had been asked to find additional effort to work on the upgrade of the PS control system. The old system had been replaced by one based on Norsk Data computers and Tim had responsibility for the user interface to the particle booster. This was a 64 character by 24 line display and Tim put together a system that showed an overview of the whole system and more detail could be brought up by clicking on certain parts. Robert Cailliau, who had joined CERN in 1974 was also working on the PS system.

During his six month stay at CERN, Tim wrote a system called ENQUIRE to help him remember all the systems, programs, things and people that he had to deal with ([Appendix B](#)).

The name ENQUIRE came from a Victorian book called **Enquire Within Upon Everything** which was first published around 1850 and by 1862 had sold 196,000 copies with almost no advertising. By 1912, it had sold 1,428,000 copies and Tim's parents had a copy. His godfather also gave him a later version.

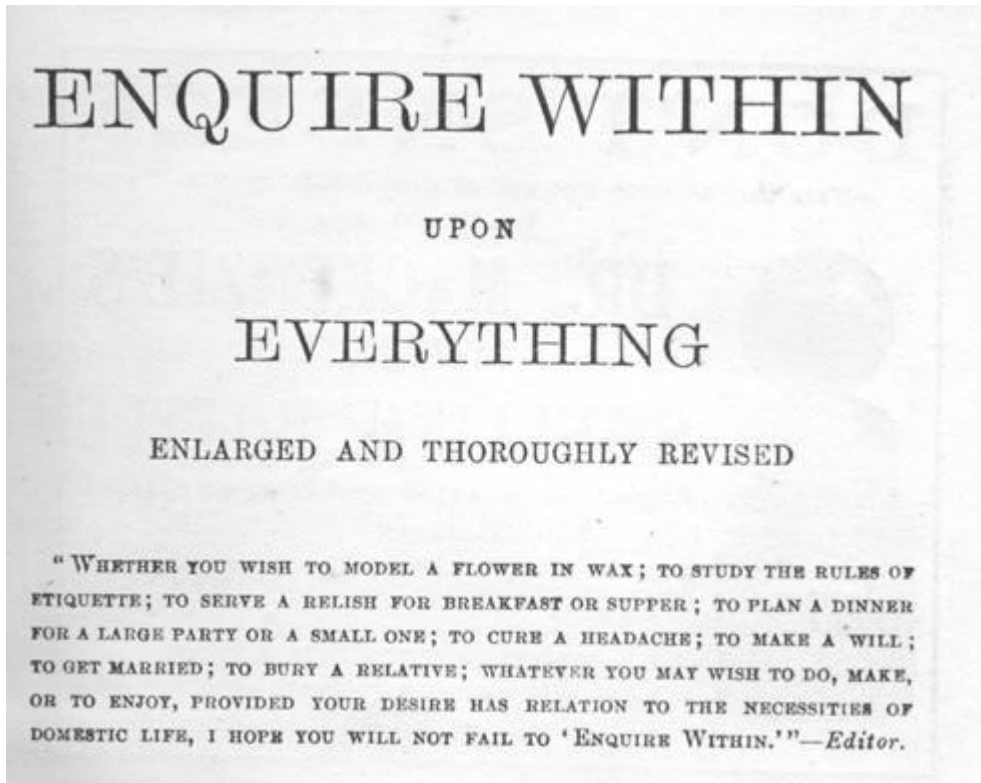


Figure 6.2: Enquire Within Upon Everything: 1912 Edition

The early editions of the book contained about 3000 short pithy descriptions of almost anything you wanted to know about. These were grouped into sections and cross-indexed. It was one of a set of 20 books in a similar style all selling at around 2s 6d each. Between them they consisted of 7000 pages of **hypertext** with about 500 pages of **links** that told you all you needed to know on any subject. Here is an example entry from the 1862 edition:

2256. Hanging

Loosen the cord, or whatever suspended the person, and proceed as for drowning, taking the additional precaution to apply eight or ten leeches to the temples.

By the 1912 edition that had become:

754. Hanging

Loosen the cord, or whatever it may be by which the person has been suspended. Open the temporal artery or jugular vein, or bleed from the arm; employ electricity, if at hand, and proceed as for drowning, taking the additional precaution to apply eight or ten leeches to the temples.

Hanging does not appear in the 1967 edition! Obviously less of a routine occurrence by then.

Tim remembered the little aphorisms that headed each page: **never buy what you do not want because it is cheap** and **knowledge is the wing whereby we fly to heaven**. Tim's ENQUIRE effectively allowed users to add nodes to a hypertext system. An interesting property was that you could not add a node unless it was linked to something else.

In his book [\[1\]](#), Tim states:

*In Enquire, I could type in a page of information about a person, a device, or a program. Each page was a **node** in the program, a little like an index card. The only way to create a new node was to make a link from an old node. The links from and to a node would show up as a numbered list at the bottom of each page, much like the list of references at the end of an academic paper. The only way of finding information was browsing from the start page.*

6.3 Second Period at CERN

Tim had been impressed by CERN and when the opportunity came up to work on the new LEP system, he applied for and got a CERN Fellowship and arrived back in September 1984. Initially working for Peggie Rimmer, one of his first jobs was to write a **Remote Procedure Call (RPC)** program to allow the varied systems at CERN to interwork. A program on one system could call up routines on other systems. This was very necessary at CERN where individual parts of the system were being produced by the experimental team all over the world on different computer systems.

Tim thought about reimplementing ENQUIRE but wanted to get away from the major constraint it had which was that all the files had to be on one machine. The work on RPC gave him the insight into how to do this. By 1988, Tim was wanting to put a proposal into CERN for a real world-wide hypertext system. At that time Ben Segal [11] was pushing the benefits of the Internet as a way of linking systems. Tim was convinced and even implemented the RPC to work over TCP/IP. CERN needed a documentation system that worked across the 250-people teams and the one that ran on the mainframe computers would not do the job with the proliferation of systems that people worked on so it would have to be distributed across many platforms. Linking documents together was not a problem. ENQUIRE already had done that. Robert Cailliau [2] was extolling the virtues of HyperCard. So having things that you could click on was a tried approach that worked. Marking up the information was relatively straightforward. The SGML gurus at CERN could help there. The one thing missing was a simple but common addressing scheme that worked between files world-wide. Tim put a proposal into CERN management via his boss Mike Sendall in March 1989 ([Appendix C](#)) but got no response from the management. He continued getting up to speed on the Internet and hypertext, now that he knew what it was called. Tim visited the USA in early 1990 and attended a workshop on hypertext in Gaithersburg, Maryland. A NeXT machine had arrived at CERN and Tim was most impressed. In March 1990 he went with Ben Segal to Mike Sendall to ask if they could buy one. Mike Sendall suggested he should buy one and try it out by developing a pilot implementation of the system in the proposal. The proposal was reformatted, a different date added, May 1990, and resubmitted. It was called this the WorldWideWeb, no spaces in those days.

The proposal clearly shows the link to the earlier ENQUIRE program. The links are described as having well defined properties as in a knowledge engineering system. Not much is said on the three main problems:

- What will documents look like?
- How will they be transmitted?
- How will they be addressed?

Peggie Rimmer said 'I've looked repeatedly at that March paper and I couldn't honestly tell you even now that it's about the World Wide Web but somehow Mike [Sendall] could.'

The NeXT arrived in September 1990, mainly due to the slowness of CERN's ordering system. By Christmas Tim had a working prototype browser and server running on the NeXT computer (Figure 6.3). Clicking on a link always opened a new window. Another major difference from most current browsers was that it was just as easy to edit pages as to view them. The server went live on Christmas Day 1990.

Robert Cailliau joined forces with Tim to put a more detailed proposal to CERN management in November 1990 (Appendix D); the term **WorldWideWeb** was used. It asked for 5 people to work on a project for 6 months and CERN would get a universal information management system. Management supported the proposal but at a lower level. Tim and Robert could work on the project plus Nicola Pellow who would write the simple browser that would run on any machine, the line mode browser (see Figure 6.4). The project had enough effort to get started but would not move forward as quickly as people would have liked.

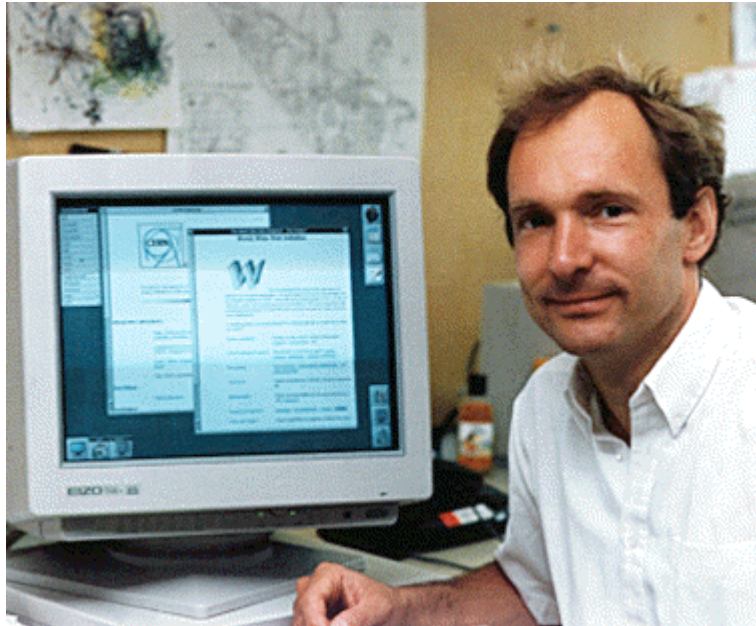


Figure 6.3: Tim's Browser for the NeXT

HTML had been agreed as the document markup language. This was based on the application of SGML used at CERN called GUID. Tim included in HTML tags from the SGML tagset used at CERN. The initial HTML parser ignored tags which it did not understand, and ignored attributes which it did not understand from the CERN-SGML tags. As a result, existing SGML documents could be made HTML documents by changing the filename from `xxx.sgml` to `xxx.html`. The earliest known HTML document is dated 3 December 1990:

```
<h1>Standardisation</h1>
There was not a lot of discussion of this at <a
href=Introduction.html>ECHT90</a>, but there seem to be two leads:
<ol>
<li><a href=People.html#newcombe>Steve newcombe's</a> and Goldfarber's
"Hytime" committee
looking into SGML, and
<li>An ISO working group known as MHEG, "Multimedia/HyperText Expert
Group".
led by one Francis Kretz (Thompsa SA? Rennes?).
</li>
```

Note the error on the last line. If you look at documents from the period between early December and the end of the year, it is clear that SGML documents have been copied across without much change relying on the fact that illegal tags would be ignored. Some of the tags in the early documents in December 1990 that did not appear in HTML include:

```
<H1>Bibliography</H1>
<BL>
<BIB ID=RPCIMP>
T.J. Berners-Lee CERN/DD,
"RPC Internals" version 2.3.1, formerly "RPC Implementation Guide"
. . . . .
To refer to a package MYPACK2 on which is provided by that server, the
client side would specify
<xmp>
MYPACK2@FRED.DRB
</xmp>
. . .
<box>
Example:
<xmp>
AA_00_04_00_2F_58_5050.ETHERNET
</xmp>
This refers to ethernet node AA_00_04_00_2F_58 (which might be a
particular VAX, for example) and protocol type 5050.
</box>
. . .
<H2>M680X0 Run Time Support under MoniCa or RMS68K

<I1 IX=1>M680x0, Run Time Support
Software exists to support RPC on the M68000/M68020 family,
in a single task environment or an RMS68k multitask environment.

The runtime library modules to be used will depend on the system.
If the RPC library is already burnt into ROM, applications
may be linked directly to the stubs and the rom entry point file.

Application software may be in C, Pascal, FORTRAN or PILS,
with stubs generated in styles GENERICC, CERNCROSS, FORTRAN or PILS.

On the Valet-Plus
<fn>This is a standard test system for High Energy Physics
built in VME equipment practice
</fn>
```

The major change is that the anchor tag does appear linking the documents together. The tag set evolved over the next year or so. There was discussion over whether the document should be a linear sequence or highly structured.

Files were accessed via the Universal Document Identifier (UDI) built on the existing Domain Name System (DNS) developed 20 years earlier by Paul Mockapeteris. UDI later became Uniform Resource Locator (URL). This was agreed by IETF in 1992. They did not like the level of grandeur associated with **universal** and it clearly had a wider use than just documents. This was all in place by May 1991.

The initial proposal was to have a naming scheme made up of the following parts:

```
scheme : // host.domain:port / path / path # anchor
```

The **scheme** defined the protocol to be used to make the access. Possibilities are **file**, **news**, **http**, **telnet**, **gopher**, **wais** and **x500**. The **host.domain:port** consisted of the host name or its IP address followed optionally by the port number which would default to 80. The **path** defines the position in the filestore hierarchy of the document and the **#anchor** defines an anchor position within the document. If the document specified was an index it was possible to search the index as follows:

```
http://cernvm/FIND/?sgml+cms
```

The directory FIND is an index and a search is made using the keywords **sgml** and **cms**.

HTTP was the transfer protocol used. The protocol used the normal internet-style telnet protocol style on a TCP-IP link. The client made a TCP-IP connection to the host using the domain name or IP number, and the port number given in the address. If the port number was not specified, 80 was always assumed for HTTP. The server accepts the connection and the response to a simple GET request was a message in HTML, a byte stream of ASCII characters. Lines were delimited by an optional carriage return followed by a mandatory line feed character. Lines could be of any length. Well-behaved servers restricted line lengths to 80 characters excluding the CR LF pair. The message was terminated by the closing of the connection by the server. Error responses were supplied in human readable text in HTML syntax. There was no way to distinguish an error response from a satisfactory response except for the content of the text. The TCP-IP connection was broken by the server when the whole document had been transferred. The protocol was stateless, in that no state was kept by the server on behalf of the client. That is all there was in HTTP in the early days.

6.4 Early Years

Progress was made but slower than all would have liked. By making a browser toolset available, it was possible to get effort from other sites and in 1991 browsers became available for the X-Workstation, Macintosh, and PC. By the end of 1991 12 servers were up and running in Physics Departments like Graz and the Stanford Linear Accelerator Center (SLAC) in the USA. Pei Wei had an X-browser called Viola running by January 1992 (Figure 6.5).

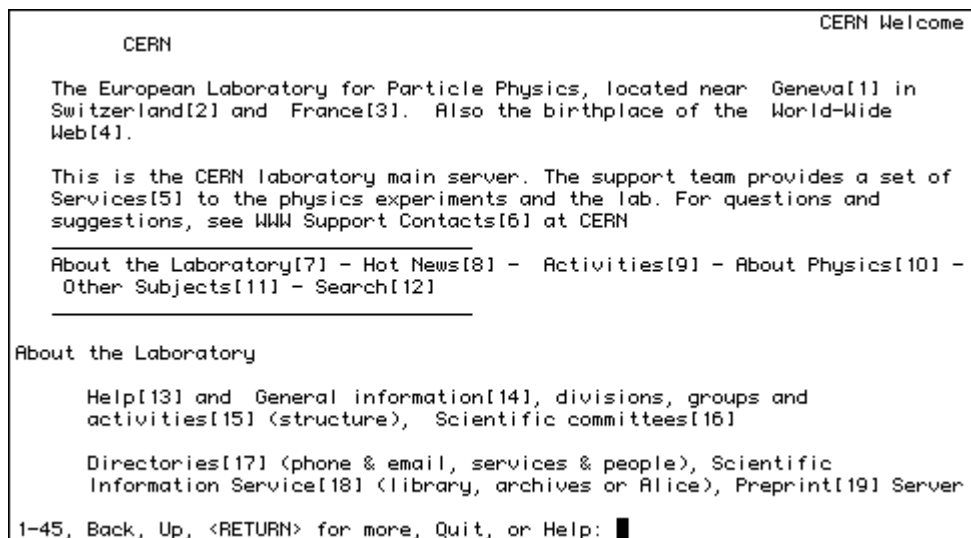


Figure 6.4: Nicola Pellow's Line Mode Browser



Figure 6.5: Pei Wei's Viola Browser

Tim and Robert put in a paper to Hypertext '91 in San Antonio that was rejected but they were allowed to demonstrate the system. The hypertext community were unimpressed. The system was far too simple for them and had no bells and whistles. It broke the commandment that links never broke. The hypertext community insisted that links never pointed to non-existent documents. That is why there systems were locked into a single database with a lot of software ensuring consistency. With the WorldWideWeb if a system went down or somebody deleted a page, an error was returned. This was sacrilege for the hypertext community.

1992 saw some progress. The number of servers world-wide grew to 50. Leeds University had a web server running and organised an Internet Day in December 1992. The Web was off the ground and developments on the Web were now as likely to be outside CERN as inside.

7. World Wide Web

- [7.1 The Web Arrives](#)
- [7.2 Web Growth](#)
- [7.3 E-Commerce](#)
- [7.4 Countries on the Web](#)

7.1 The Web Arrives

After the initial period of activity at CERN, the real question was would this take off. After nearly two years, having 50 web sites could be regarded as successful in a Particle Physics view but not from a world view.

The 1993 year saw the arrival of Illinois and Mosaic. The National Centre for Supercomputing Applications had been established in 1985 at the University of Illinois. Ping Fu had joined NCSA in 1990 to do 3D scientific visualisation. Her husband had connections with the people in Graz (Maurer had visited them in Illinois) and they had pointed her at the browser as a thing of the future. She had recently hired Marc Andreessen and suggested that he implemented a web browser. Marc was unimpressed. First he had never heard of a browser and when he worked out what it was decided it was little more than a pretty FTP system. Joseph Hardin at NCSA thought there was more potential in the browser particularly with regard to cooperative working. As a result, Marc got interested as long as the browser did a bit more than the current browsers did. He developed the Mosaic browser with others which had the additional feature that images could be added to web pages unlike Tim's NeXT browser that had them pop up in a separate window.

NCSA heard about the Web on the 9 November 1992 and had the first X-Mosaic browser running by 23 January 1993. It was added to the browser that you could download from CERN and had the distinct advantage that it was a single file and very easy to install. There were 145 downloads before the end of January, 1161 in February. Early versions of Mosaic allowed you to annotate Web pages giving the cooperative working that Joseph Hardin was interested in. Joseph Hardin put people to work on producing both PC (Chris Wilson and Jon Mittelhauser) and Mac (Aleks Totic) versions. All the versions had been released by November 1993. By October of 1993, the Web had grown to being 1% of all Internet traffic and by the end of December 1993 nearly 2.5%. The Web had taken off through the influence of Mosaic but in the process had lost the browser/editor concept of the original NeXT browser.

7.2 Web Growth

Some statistics from January 1994 show the colossal growth:

- 2 million users, 150,000 new users a month
- 95% used the Mosaic browser, 2% used the line mode browser
- 90% used Unix
- 1.5 Million hits a week at NCSA
- 11% growth a week

The Web was now unstoppable but still heavily in the academic sector. The First World Wide Web Conference was held at CERN in May 1994 and was largely a coming together of the people who had been influential in getting the Web off the ground and meeting together for the first time.

However, the commercial possibilities were becoming evident particularly with regard to the internal management of a company's information. Digital Equipment Corporation installed its first Web server in May 1993 and soon grew to 60 servers and 40,000 internal users via a great deal of effort in translating old material into formats accessible on the Web.

1995 saw the arrival of Jim Clarke's company Netscape and Microsoft woke up to the Web and produced the first version of Internet Explorer. Figure 7.1 shows the enormous growth created by having commercial browser with lots of bells and whistles available free to the community as a whole. No longer was the Web confined to the academic community.

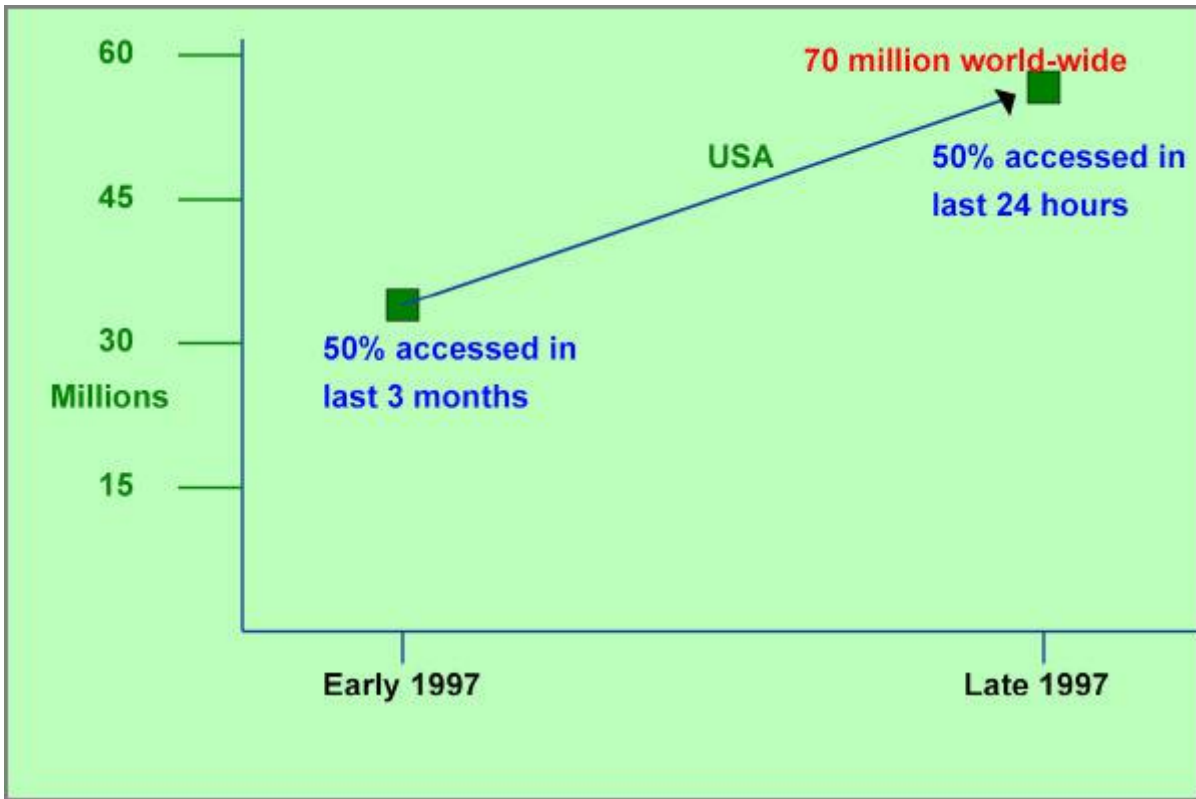


Figure 7.1: Web Statistics: 1997

While 1997 saw this enormous growth, people that were trying out the Web at the beginning of the year had by the end of the year made it part of everyday life. Figure 7.2 shows the continuing growth over the next few years. Today most people have heard of the Web, the man in the street uses it regularly and some companies base their whole future on it.

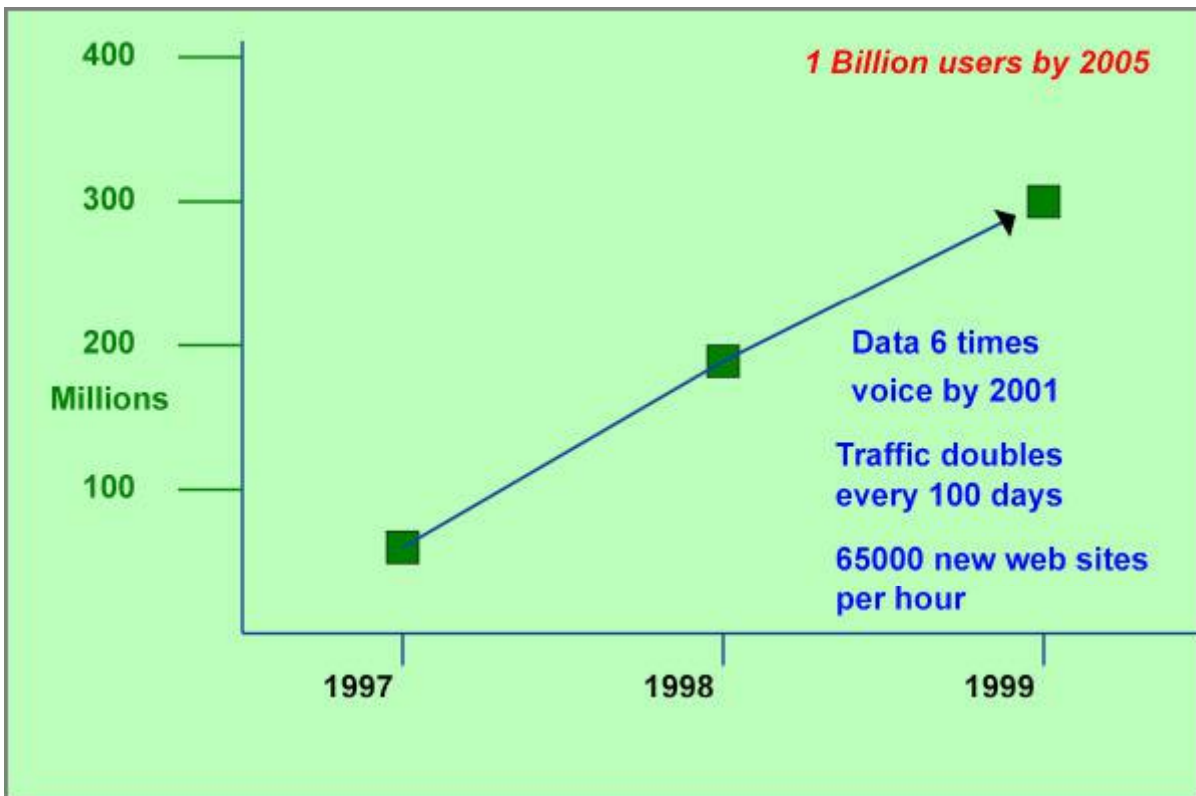


Figure 7.2: Web Statistics: 1997-1999

7.3 E-Commerce

Using the Web for commerce started in earnest in 1997. Figure 7.3 shows the rise and type of purchases being made. The worrying statistic was the relative low take up in Europe. As a result, the European Union invested significant funds into projects, WebCore and W3C-LA in the period 1997 to 2000 to try and raise awareness of the Web in Europe.



Figure 7.3: Commercial Use of the Web

The top 20 e-commerce companies around 1999 were:

1. amazon.com: books but branching out into CDs etc
2. buy.com: computer and office products
3. barnesandnoble.com: books
4. ticketmaster.com: theatre tickets
5. planetrx.com: prescription drugs now deceased
6. mothernature.com: nature products now deceased
7. drugstore.com: prescription drugs and general chemists products
8. gateway.com: computer and office products
9. cdnow.com: CDs
10. smarterkids.com: children's toys
11. chipshot.com: golf equipment
12. hallmark.com: ornaments, cards etc
13. egghead.com: computer and office products
14. yahoo.com: search engine etc
15. officemax.com: office products
16. etoys.com: children's toys
17. jcrew.com: clothing
18. spree.com: portal to all the other commercial suppliers
19. compaq.com: computer equipment
20. towerrecords.com: CDs

Only two of the 20 have stopped trading. By 1998, the Web was beginning to have an effect on the global economy. The predicted growth in the US economy was anticipated as 2.5% and there was a belief that the economy would go into recession. The actual growth was 4% largely due to ecommerce. Web commerce doubled during the year (prediction was 50%). The competitive pricing of web products lowered inflation by 1.1% and productivity grew by 3.4% largely as a result of the Web.

7.4 Countries on the Web

By the end of 1999, the top countries in terms of web usage were:

Top 15 Countries in Internet Use at Year-End 1999				
Rank	Country	Internet Users (000)	Population (000)	Percentage
1.	United States	110,825	272,878	40.6
2.	Japan	18,156	126,182	14.4
3.	UK	13,975	59,113	23.6
4.	Canada	13,277	31,006	23.6
5.	Germany	12,285	82,087	15.0
6.	Australia	6,837	18,700	36.6
7.	Brazil	6,790	171,853	4.0
8.	China	6,308	1246,872	0.5
9.	France	5,696	58,978	9.7
10.	South Korea	5,688	46,884	12.1
11.	Taiwan	4,790	22,113	21.7
12.	Italy	4,745	56,735	8.4
13.	Sweden	3,950	8,911	44.3
14.	Netherlands	2,933	15,807	18.6
15.	Spain	2,905	39,167	7.4

Source: Computer Industry Almanac

The position of other European countries was:

Country	Internet Users (000)	Population (000)	Percentage
Finland	1,430	5,158	27.7
Belgium	1,400	10,182	13.7
Norway	1,340	4,438	30.2
Switzerland	1,179	7,275	16.2
Denmark	741	5,356	13.8
Israel	500	5,749	8.7
Ireland	370	3,632	10.2
Austria	362	8,139	4.4
Portugal	200	9,918	2.0

If we look at the statistics in terms of penetration in Europe:

Country	Internet Users (000)	Population (000)	Percentage
Sweden	3,950	8,911	44.3
Norway	1,340	4,438	30.2
Finland	1,430	5,158	27.7
UK	13,975	59,113	23.6
Netherlands	2,933	15,807	18.6
Switzerland	1,179	7,275	16.2
Germany	12,285	82,087	15.0
Denmark	741	5,356	13.8
Belgium	1,400	10,182	13.7
Ireland	370	3,632	10.2
France	5,696	58,978	9.7
Israel	500	5,749	8.7
Italy	4,745	56,735	8.4
Spain	2,905	39,167	7.4
Austria	362	8,139	4.4
Portugal	200	9,918	2.0

8. World Wide Web Consortium

- [8.1 Introduction](#)
- [8.2 World Wide Web Consortium](#)
- [8.3 How W3C Works](#)

8.1 Introduction

In Section 7.3, it was mentioned that as early as 1993 companies were beginning to use the Web as a central part of their companies future strategy. The number of hits on the CERN server, info.cern.ch had grown from 100 a day in the summer of 1991 to 1,000 a day by the summer of 1992 and 10,000 a day in the summer of 1993. By 1994, things were beginning to fragment. With no real authority other than CERN in charge of Web specifications, people were beginning to add extensions all over the place. The situation got even worse once Netscape and Microsoft got involved. There needed to be a body that agreed what was HTML and HTTP or the whole thing would fragment into a set of independent ghettos and the vision of a single World Word Web would be lost.

Early in 1994, Tim discussed with Michael Dertouzos of MIT the possibility of starting a Consortium, rather like the X Consortium that had been hosted at MIT and was responsible for the X-Window developments. Tim discussed with CERN Management whether they wanted to retain control of the Web or let it go. The enormous take up in the USA compared with Europe made it unlikely that a regulatory body based in Europe would be successful. A decision was needed no later than the end of 1993. In January, 1994, Robert Cailliau put a proposal called **Alexandria**, after the great library of 300 BC, to the European Union seeking funds to keep Europe at the forefront of Web developments. For 2.84 MECU, Europe would have a centre of 32 people defining and maintaining the standards to be used on the Web. After much discussion between CERN, MIT and the European Union, there was a draft agreement to set up a Consortium with the USA arm at MIT (with Tim moving to Boston) and the European arm at CERN. Funding for the European arm would come from the European Union. In July, MIT announced the new Consortium based at MIT and CERN and said it would formally start in October. Companies were invited to join. This preemptive announcement by MIT did not help negotiations. There was much discussion over whether the European and US arms were of equal weight etc. Cutting a long story short, CERN eventually decided to pull out of the agreement after a meeting in December 1994. In consequence, The European Union asked INRIA in France to become the European arm which they accepted.

8.2 World Wide Web Consortium

The World Wide web Consortium (W3C) was set up **to realise the full potential of the Web as a robust scaleable adaptive infrastructure preserving interoperability**. INRIA was a member of the European Research Consortium for Informatics and Mathematics (ERCIM) which was a Consortium of the major IT Government-funded Research Laboratories in Europe. The UK representative was RAL (another Particle Physics Laboratory), the German representative was GMD and for the Netherlands there was CWI. In all there were 10 members of ERCIM. These rallied round to help INRIA establish the European arm of W3C. Jean-Francois Abramatic became the manager of the European Office of W3C and his staff, situated at Sophia Antipolis near Nice, comprised a number of the CERN staff early on plus some support from the ERCIM members. At MIT, the W3C Manager was Al Vezza who had been responsible for the X-Windows Consortium. Tim was the Director, now resident at MIT. Funding for the European Office came initially from a EU Contract called WebCore that established the W3C Office at INRIA. To improve the European involvement, this was followed in the period 1997-1999 by the European Leveraging Action (W3C-LA) that established a set of European Offices of W3C at the ERCIM partners organised by Bob Hopgood at RAL.

W3C from an initial membership of about 50 companies has grown to well over 500 and continues to be the organisation that defines the specifications that are used on the Web and that its members are committed to implement. In the USA, companies like Microsoft, IBM, Sun, Hewlett Packard, Apple, Adobe and Boeing are members. In the UK, membership includes BT, Oxford Brookes, BBC, Sema, APACS, ECA, RAL, Southampton, Bristol, Edinburgh Univ, RivCom, AND-Data, Brunel, etc. In Europe, Philips, Nokia, Ericsson, Reuters, Siemens etc are members. In Asia, Fujitsu, NEC, Matsushita, Mitsubishi, Hitachi, Honda etc are members.

The W3C Membership currently (July 2001) stands at:

Type	Americas	Europe	Pacific	Total
Full	62	29	17	108
Affiliate	240	123	47	410
Host	1	1	1	3
Total	303	153	65	520

8.3 How W3C Works

The activities of W3C fall into the following categories:

- Recommendations
 - Members support for industry-wide adoption
 - Web standards
- Sample code
 - Jigsaw server
 - Amaya browser
 - Tools library
- Assistance
 - Validators
 - Design guides

Recommendations are the standards of the Web. The initial ones were HTML and HTTP. The list is now well over 50 standards. W3C adopts the IETF philosophy of having sample code for all the major innovations and W3C does this via a server called Jigsaw and a browser called Amaya. Associated with the standards are a set of additional products that include validators to ensure that they are used correctly and guides to use them in the way they were intended.

The W3C process is as follows:

- Member(s) **submit** a request for activity in specific area
- Workshops to define need
- Working Group to achieve consensus
- Proof of Concept implementations
- Working Draft
- Candidate Recommendation
- Proposed Recommendation
- Voting
- **W3C Recommendation (Standard)**
- Reference Code (Jigsaw Server, Amaya Browser)
- Validators etc

The W3C has moved the Web forward since 1994 in three major stages:

- **Consolidation:** getting the basic protocols back to a state where interworking was again possible.
- **Evolution:** improving the separation of style and content, adding functionality and looking at societal needs particularly in the areas of accessibility and child protection.
- **Revolution:** the radical changes currently underway to move the focus from HTML to XML.

In 2001, W3C is recognised as the controlling body for the Web. Its membership are active in moving the Web forward. The immediate future is secure and what started as one person's attempt to provide a system that emulated a Victorian book is now the major focus for information storage and retrieval world-wide.

Appendix A

References

1. Weaving the Web, Tim Berners-Lee with Mark Fischetti, Harper, 1999.
2. How the Web was Born, James Gillies and Robert Cailliau, Oxford University Press, 2000.
3. Netscape Time, Jim Clark, St martin's Press, 1999
4. The New New Thing, Michael Lewis, Norton, 2000
5. The Victorian Internet, Tom Standage, Weidenfeld and Nicholson, 1998.
6. The SGML Handbook, Charles Goldfarb, Clarendon Press, 1990.
7. The Whole Internet, Ed Krol, O'Reilly, 1992.
8. Enquire Within Upon Everything, 123rd Edition Herbert Jenkins, 1967 1992.
9. <http://www.dei.isep.ipp.pt/docs/arpa-Contents.html>: History of the ARPANET
10. <http://www.isoc.org/internet/history/index.shtml>: Internet Histories
11. <http://www.isoc.org/internet/history/index.shtml>: A Short History of Internet Protocols at CERN, Ben Segal
12. <http://www.sgmlsource.com/history/index.htm>: The SGML History Niche
13. Text Processing and Typesetting with Unix, David Barron and Mike Rees, Addison Wesley, 1987
14. <http://www.ps.uni-sb.de/~duchier/pub/vbush/vbush.shtml>: As we May Think, Vannevar Bush, Atlantic Monthly, July 1945
15. <http://www.xanadu.com.au/general/faq.html>: Xanadu FAQ
16. http://www.isg.sfu.ca/~duchier/misc/hypertext_review/index.html: Rutgers Hypertext Course
17. Hypermedia and the Web, David Lowe, Wendy Hall, Wiley. 1999.

Appendix B: ENQUIRE

Contents

- [1. Introduction](#)
- [2. Scope](#)
- [3. Modular Structure in General: Circles and Arrows](#)
- [4. Objectives](#)
 - [4.1 Documentation](#)
 - [4.2 Analysis](#)
 - [4.3 Control](#)
 - [4.4 Reminder](#)
- [5. Using ENQUIRE-WITHIN](#)
 - [5.1 Implementation under SYNTRAN III](#)
 - [5.2 Display format](#)
 - [5.3 Commands](#)
 - [5.4 Edit Mode](#)
 - [5.5 User Input](#)
 - [5.6 Languages](#)
 - [5.7 Possible extensions](#)

1. Introduction

ENQUIRE is a method of documenting a system. It concentrates on the way the system is composed of parts, and how these parts are interrelated.

This information about a system is difficult to store, and particularly to update, using paper documents, so ENQUIRE stores its information on a computer. The ENQUIRE-WITHIN program allows a person to create and edit the data, so that others can later extract it as it is relevant to them.

2. Scope

The ENQUIRE system is designed to fill a gap in many current documentation systems. A person finding himself faced with a piece **xxx** of a system should be able to ask ENQUIRE, for example

- What is **xxx** part of?
- What is **xxx** composed of?
- What must I alter if I change **xxx**?
- What facilities does **xxx** use?
- Where do I find out more about **xxx**?
- Who or what produced **xxx**?

ENQUIRE does not aim to answer such questions a

- How does **xxx** work?
- What format is **xxx** in, exactly?
- Why was **xxx** created?
- What is the format of the interface between **xxx** and **yyy**?

The answers to these questions are usually covered by the descriptions and specifications of **xxx**. ENQUIRE may be of use in helping a user to find such documents, but it does not reckon to store them.

3. Modular Structure in General: Circles and Arrows

The assumption is made that the system to be described can be broken up into **modules**. It is generally accepted that this is a necessity for any modifiable or maintainable system involving computers. No assumptions are made about how the breaking up is done - ENQUIRE imposes no constraints on the high level design.

A similar way of describing a structure is to draw on a piece of paper, circles with arrows in between. The circles (**modules**) could be programs or pieces of hardware, for instance, and the arrows could mean **passes data to, is composed of or is started by**. This method, with a variety of different shaped boxes, and different coloured arrows is useful, clear, and commonly used. The ENQUIRE system allows a more complicated system to be described than would fit on a piece of paper. It then allows an interactive user to explore the system in search of the information he requires, seeing only the parts which are of interest.

ENQUIRE divides both the modules (circles) and relationships (arrows) into broad categories. This makes it easier to analyse the structure you end up with. For instance, the relationships **is part of** and **includes** show the division of the module into smaller modules. Also, when altering one part of the system, it is useful to know by which other parts it is used. These are generalised relationships just as **Document, Program, Machine** are generalised types of module.

The modules may be all sorts of things. They are referred to below as **nodes** because of the role they take in the network of interrelationships within the system.

4. Objectives

4.1 Documentation

The primary objective of ENQUIRE is to store and retrieve information about the structure of a system. This is done with the aid of the interactive ENQUIRE-WITHIN program. As a side-effect to this, those people defining the structure of a system using ENQUIRE may find themselves thinking about it, and with luck a proliferation of interdependencies may be avoided.

4.2 Analysis

As a second stage, the data produced may be interesting to investigate. This may show

- whether the eventual structure of the system matches its design goals;
- when changes are required, what sort of changes would make the system cleaner and more manageable, rather than less so.

Whether this form of analysis is useful is yet to be determined.

4.3 Control

When a system is expanding rapidly, some control is normally required over the way it is allowed to expand. If rules are made, it may be that ENQUIRE provides a way of checking that these rules are adhered to and that aspects of the structure have not been introduced which will degrade the system in the long term. (The topic of project control for complex systems is a large one, and is not discussed here.)

4.4 Reminder

It must be emphasised that the ENQUIRE system is not intended to replace other forms of documentation, and is designed to work alongside other indexing systems and retrieval systems.

5. Using ENQUIRE-WITHIN

The following is an outline of the facilities available from the interactive editor. Note that the program assumes a VDU with a speed of at least 200 baud which accepts the backspace character.

5.1 Implementation under SYNTRAN III

The ENQUIRE program is currently stored on file (GUEST)ENQUIRE-WITHIN:PROG on the PPDEV computer on the CERN PS control system. Log in to any of the interactive VDU terminals. At SYNTRAN III command level, type

```
@(GUEST)ENQUIRE <param>
```

and the system should respond

```
Enquire V x.x  
Hello!
```

The only <param> which is acceptable at present (Oct 80) is the keyword EDIT. If this is present, you will be allowed to alter those files to which you have write access under the operating system.

The database is stored on various **continuous** files called `xxxxxxxxxx-V1:ENQR`. The 12 characters `xxxxxxxxxx` include the user name under which the file is stored, and part of the filename. Separate files are used for storing information about different parts of systems, but an object in one file may freely refer to objects in other files.

```
@ENQUIRE  
Enquire V.1.  
Hello!  
Opening file (PSK-POP)VAC-V1:ENQR...  
PSB Vacuum Control System  
-----  
[ 1] described-by: Enquire System  
      An experimental system for which this is a test.  
  
[ 2] includes: Vacuum History System           Records and displays slow  
changes in pressure.  
  
[ 3] includes: Vacuum Equipment modules  
      Perform all the hardware interfaces  
  
[ 4] includes: Control and status applications program  
      Provide operator interaction from the consoles.  
  
[ 5] described-by: Con?rcle du System a Vide du Booster 11-2-80  
      Operational specification of the software  
  
[ 6] includes: PSB Pump Surveillance System PCP 228  
      Allows rapid monitoring of pressure changes  
  
[number  ]
```

Figure 1: The PSB Vacuum Control node in the file (PSK-PCP)Vac-V1:ENQR. The [prompt] is shown on the last line.

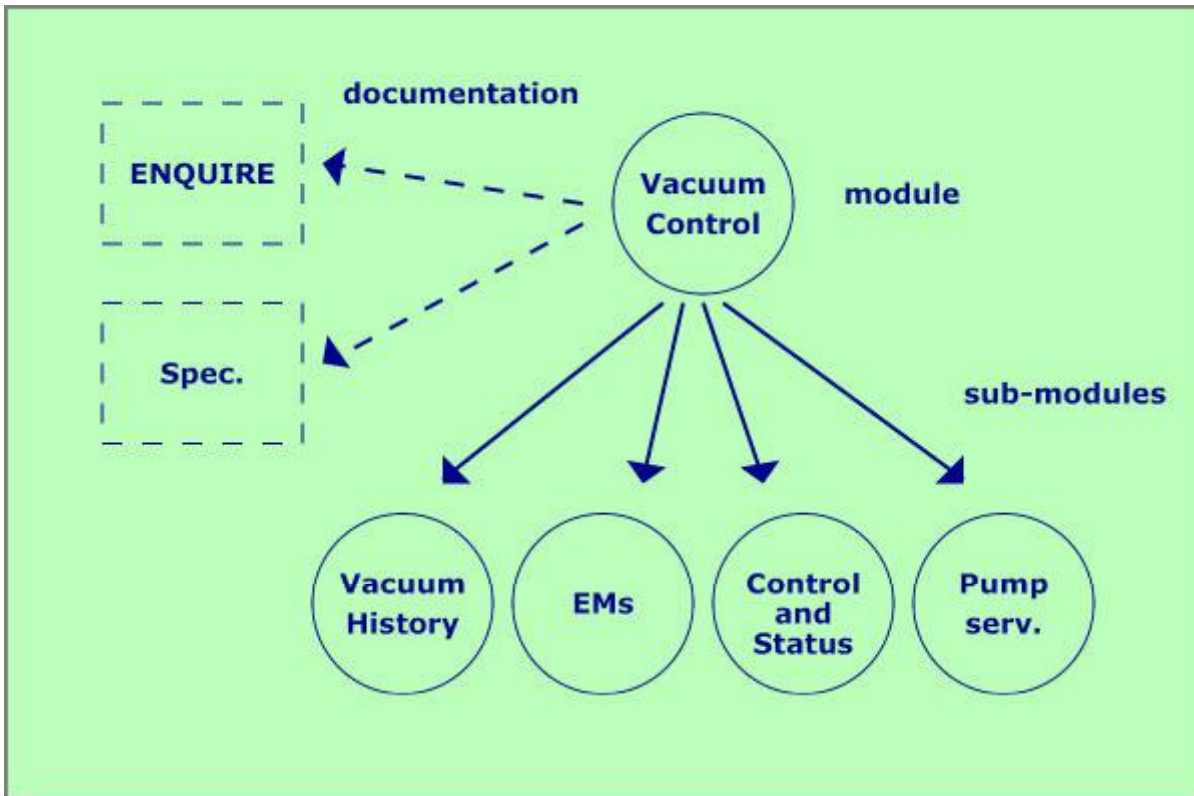


Figure 2: Shows a circles and arrows representation of the same information.

5.2 Display format

On entry into ENQUIRE, and at various points during its use, the user is presented with a page much like Figure 1. This gives information about one of the **nodes**, and the relationship between it and other nodes. The corresponding circle and arrow diagram is shown in Figure 2.

Information displayed is as follows:

For each node,

1. **Name:** Name of the node, filename if it represents a file
2. **Sort:** Category into which the thing described is placed
3. **Comment:** Any further text considered relevant by the last editor.

For each related node:

1. **Number:** Allows one of the nodes referred to to be selected
2. **Name:** The name of the related node.
3. **Relationship:** Generalised relationship between the two nodes.
4. **Comment:** A description of the node referred to. This text describes the way it is relevant to the node being listed.

The following **sorts** of nodes are currently (Oct80) chosen from the following set:

hardware

a piece of equipment

text-file

readable ASCII information stored on computer.

report-file

text information accessible by computer, best read after being processed by the REPORT program.

code-file

information stored on the computer, not human-readable.

concept

an abstract idea, anything without a physical form.

person

those strange unreliable beings on which all systems eventually depend.

The type of relationship is currently one of the following set:

uses used-by	Complementary. Reflect that one node relies on the existence of the other.
includes part-of	Complementary. One module is a sub-module to the other.
made made-by	Complementary. One module, program or person, was responsible for the other.
describes described-by	Complementary. Useful for linking documentation to that which it is relevant.
background detail	Link different documents together
similar-to other	If none of the specific relationships above hold, one of these is chosen.

5.3 Commands

On entry, the user is in the outer command level of ENQUIRE. He is prompted by two brackets, and may insert a command word between them. The following commands are available at this level to allow interrogation of the data. Commands marked * also allow alteration of the data.

1,2,3 etc.	Select one of the nodes referred to by this one. Move on to that node, and display its relevant data.
Mark	Remember this node. The user marks nodes which interest him. A list of up to 16 such marked nodes is maintained. This list is displayed after each mark command.
Unmark	The last node marked as interesting is removed from the list.
List	List all the references and comment about this node.
Quit	Leave the ENQUIRE program.
Create *	Make a new enquiry data file. the file must already exist under SYNTRAN III as a continuous file of suitable size.
Extend *	Add another node to this file
Edit *	Go into EDIT MODE, to alter information about this node.
French	Load new vocabulary (See Section 5.6)

Note: The commands marked * are only available if the user has write access to the file and gave the EDIT parameter when originally invoking the ENQUIRE program. (See Section 5.1)

5.4 Edit Mode

Edit Mode allows the user to alter the details about the current node. He is prompted for an edit mode sub-command by a different distinctive prompt:

```

    >>edit object [      ]
or
    >>edit reference [      ], etc.

```

The prompt shows what part of the data he is dealing with. The set of actions possible in EDIT mode differ from those possible at the outer command level (Section 5.3). Some of these have the same keywords as those at the outer command level, but should not be confused with them.

create	Create a new reference, from this node to one of the ones marked as interesting. The user is prompted with a list of such nodes, and selects one. he then supplies a few details about the node selected, from the particular point of view of the node he is editing.
list	List the bit of data I am currently dealing with.
edit	Edit the bit of data I am currently dealing with.
extend	Add a new bit of comment immediately after this bit of data. If there is already some comment tagged on, more may only be added to the end of the last line.
	The following sub-commands affect which bit of data is to be dealt with next.
object	Deal with the name of the node and its sort.
reference	Deal with the next reference.
comment	Deal with the next line of comment.
quit	Leave edit mode, return to the outer command level of ENQUIRE

5.5 User Input

After giving a command or sub-command, the user may be prompted for further input. The prompts should be self-explanatory. when entering any information, the following control characters may be used.

Control/D	means accept the line as it is currently displayed
Return	means accept the line to the left of the cursor unless the cursor is under the first character of the line, in which case the meaning is the same as for Control/D. Answering return to a question immediately, therefore, will give the default answer if one exists. also,
Backspace	moves back (left) one character (also Control/H)
Delete	means delete last character (also Control/A)
Control/S	means delete next character
Control/Q	means delete line (Also Control/U)
Control/Y	means move to beginning of the line
Control/B	means move to end of the line

When a keyword is required,

1. a default keyword is always available, if Control/D is typed
2. entering garbage will produce a list of keywords which are valid in context;
3. UPER and lowercase letters are treated identically.
4. Any abbreviation may be used which is unique in context.

Commands, sorts of nodes, types of relationship are examples of keywords.

5.6 Languages

The vocabulary used by ENQUIRE, and the keywords it accepts as input, are loaded from a special file on disk. It is therefore possible to alter the language the program uses, or the one it communicates in. This is only of limited effect, as the program cannot unfortunately, translate the entire database into a different language.

To use a different vocabulary, use one of the allowed language names as a command at the outer level of ENQUIRE.

Example:

```
[French    ]  
Bonjour!  
  
[Anglais  ]  
Hello!  
  
(User input is between [Brackets] )
```

5.7 Possible extensions

The following are extensions which could be provided to the system:

- Sophisticated key searches of node names
- Automatic generation of index files from ENQUIRE data
- Automatic tracing of all dependent modules, component modules, or required modules for any node;
- Various forms of hard-copy printout
- A file tidying and garbage collecting facility for shortening files;
- Global editing facilities for making many related changes to a file or set of files.

Appendix C: Proposal to CERN Management, March 1989 and May 1990

A hand conversion to HTML of the original MacWord (or Word for Mac?) document written in March 1989 and later redistributed unchanged apart from the date added in May 1990. Provided for historical interest only. The diagrams are a bit dotted, but available in versions linked below. The text has not been changed, even to correct errors such as misnumbered figures or unfinished references.

This document was an attempt to persuade CERN management that a global hypertext system was in CERN's interests. Note that the only name I had for it at this time was "Mesh" -- I decided on "World Wide Web" when writing the code in 1990.

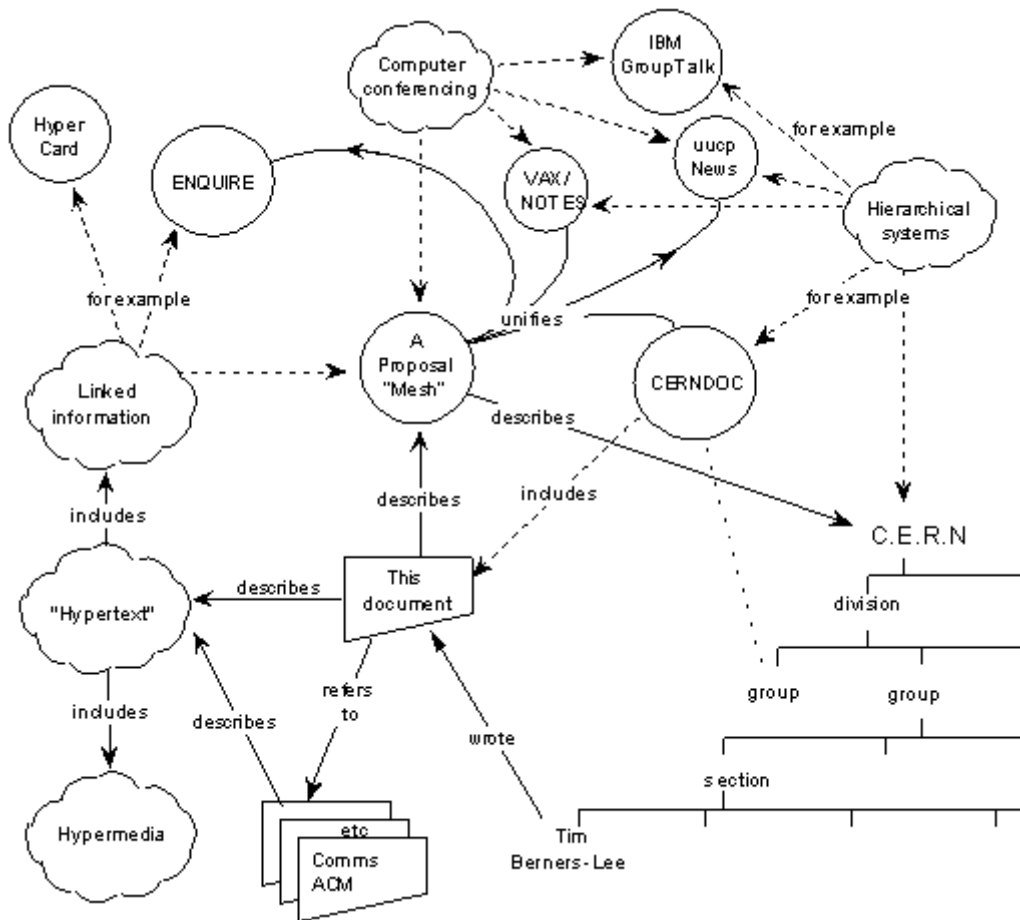
©Tim Berners-Lee 1989, 1990, 1996, 1998. All rights reserved.

Information Management: A Proposal

Tim Berners-Lee, CERN

March 1989, May 1990

This proposal concerns the management of general information about accelerators and experiments at CERN. It discusses the problems of loss of information about complex evolving systems and derives a solution based on a distributed hypertext system.



Overview

Many of the discussions of the future at CERN and the LHC era end with the question - "Yes, but how will we ever keep track of such a large project?" This proposal provides an answer to such questions. Firstly, it discusses the problem of information access at CERN. Then, it introduces the idea of linked information systems, and compares them with less flexible ways of finding information.

It then summarises my short experience with non-linear text systems known as **hypertext**, describes what CERN needs from such a system, and what industry may provide. Finally, it suggests steps we should take to involve ourselves with hypertext now, so that individually and collectively we may understand what we are creating.

Losing Information at CERN

CERN is a wonderful organisation. It involves several thousand people, many of them very creative, all working toward common goals. Although they are nominally organised into a hierarchical management structure, this does not constrain the way people will communicate, and share information, equipment and software across groups.

The actual observed working structure of the organisation is a multiply connected "web" whose interconnections evolve with time. In this environment, a new person arriving, or someone taking on a new task, is normally given a few hints as to who would be useful people to talk to. Information about what facilities exist and how to find out about them travels in the corridor gossip and occasional newsletters, and the details about what is required to be done spread in a similar way. All things considered, the result is remarkably successful, despite occasional misunderstandings and duplicated effort.

A problem, however, is the high turnover of people. When two years is a typical length of stay, information is constantly being lost. The introduction of the new people demands a fair amount of their time and that of others before they have any idea of what goes on. The technical details of past projects are sometimes lost forever, or only recovered after a detective investigation in an emergency. Often, the information has been recorded, it just cannot be found.

If a CERN experiment were a static once-only development, all the information could be written in a big book. As it is, CERN is constantly changing as new ideas are produced, as new technology becomes available, and in order to get around unforeseen technical problems. When a change is necessary, it normally affects only a small part of the organisation. A local reason arises for changing a part of the experiment or detector. At this point, one has to dig around to find out what other parts and people will be affected. Keeping a book up to date becomes impractical, and the structure of the book needs to be constantly revised.

The sort of information we are discussing answers, for example, questions like

- Where is this module used?
- Who wrote this code? Where does he work?
- What documents exist about that concept?
- Which laboratories are included in that project?
- Which systems depend on this device?
- What documents refer to this one?

The problems of information loss may be particularly acute at CERN, but in this case (as in certain others), CERN is a model in miniature of the rest of world in a few years time. CERN meets now some problems which the rest of the world will have to face soon. In 10 years, there may be many commercial solutions to the problems above, while today we need something to allow us to continue.

Linked information systems

In providing a system for manipulating this sort of information, the hope would be to allow a pool of information to develop which could grow and evolve with the organisation and the projects it describes. For this to be possible, the method of storage must not place its own restraints on the information. This is why a "web" of notes with links (like references) between them is far more useful than a fixed hierarchical system. When describing a complex system, many people resort to diagrams with circles and arrows. Circles and arrows leave one free to describe the interrelationships between things in a way that tables, for example, do not. The system we need is like a diagram of circles and arrows, where circles and arrows can stand for anything.

We can call the circles nodes, and the arrows links. Suppose each node is like a small note, summary article, or comment. I'm not over concerned here with whether it has text or graphics or both. Ideally, it represents or describes one particular person or object. Examples of nodes can be

- People
- Software modules
- Groups of people
- Projects
- Concepts
- Documents
- Types of hardware
- Specific hardware objects

The arrows which links circle A to circle B can mean, for example, that A...

- depends on B
- is part of B
- made B
- refers to B
- uses B
- is an example of B

These circles and arrows, nodes and links, have different significance in various sorts of conventional diagrams:

Diagram	Nodes are	Arrows mean
Family tree	People	"Is parent of"
Dataflow diagram	Software modules"	Passes data to"
Dependency	Module	"Depends on"
PERT chart	Tasks	"Must be done before"
Organisational chart	People	"Reports to"

The system must allow any sort of information to be entered. Another person must be able to find the information, sometimes without knowing what he is looking for.

In practice, it is useful for the system to be aware of the generic types of the links between items (dependences, for example), and the types of nodes (people, things, documents..) without imposing any limitations.

The problem with trees

Many systems are organised hierarchically. The CERNDOC documentation system is an example, as is the Unix file system, and the VMS/HELP system. A tree has the practical advantage of giving every node a unique name. However, it does not allow the system to model the real world. For example, in a hierarchical HELP system such as VMS/HELP, one often gets to a leaf on a tree such as

```
HELP COMPILER SOURCE_FORMAT PRAGMAS DEFAULTS
```

only to find a reference to another leaf: "Please see

```
HELP COMPILER COMMAND OPTIONS DEFAULTS PRAGMAS"
```

and it is necessary to leave the system and re-enter it. What was needed was a link from one node to another, because in this case the information was not naturally organised into a tree.

Another example of a tree-structured system is the uucp News system (try 'rn' under Unix). This is a hierarchical system of discussions ("newsgroups") each containing articles contributed by many people. It is a very useful method of pooling expertise, but suffers from the inflexibility of a tree. Typically, a discussion under one newsgroup will develop into a different topic, at which point it ought to be in a different part of the tree. (See Fig 1).

```
From mcvox!uunet!pyrdc!pyrnj!rutgers!bellcore!geppetto!duncan Thu Mar...
Article 93 of alt.hypertext:
Path: cernvox!mcvox!uunet!pyrdc!pyrnj!rutgers!bellcore!geppetto!duncan
>From: duncan@geppetto.ctt.bellcore.com (Scott Duncan)
Newsgroups: alt.hypertext
Subject: Re: Threat to free information networks
Message-ID: <14646@bellcore.bellcore.com>
Date: 10 Mar 89 21:00:44 GMT
References: <1784.2416BB47@isishq.FIDONET.ORG> <3437@uhccux.uhcc...
Sender: news@bellcore.bellcore.com
Reply-To: duncan@ctt.bellcore.com (Scott Duncan)
Organization: Computer Technology Transfer, Bellcore
Lines: 18
```

Doug Thompson has written what I felt was a thoughtful article on censorship -- my acceptance or rejection of its points is not particularly germane to this posting, however.

In reply Greg Lee has somewhat tersely objected.

My question (and reason for this posting) is to ask where we might logically take this subject for more discussion. Somehow alt.hypertext does not seem to be the proper place.

Would people feel it appropriate to move to alt.individualism or even one of the soc groups. I am not so much concerned with the specific issue of censorship of rec.humor.funny, but the views presented in Greg's article.

Speaking only for myself, of course, I am...

Scott P. Duncan (duncan@ctt.bellcore.com OR ...!bellcore!ctt!duncan)
(Bellcore, 444 Hoes Lane RRC 1H-210, Piscataway, NJ...)
(201-699-3910 (w) 201-463-3683 (h))

Fig 1. An article in the UUCP News scheme.

The Subject field allows notes on the same topic to be linked together within a "newsgroup". The name of the newsgroup (alt.hypertext) is a hierarchical name. This particular note expresses a problem with the strict tree structure of the scheme: this discussion is related to several areas. Note that the "References", "From" and "Subject" fields can all be used to generate links.

A solution: Hypertext

The problem with keywords

Keywords are a common method of accessing data for which one does not have the exact coordinates. The usual problem with keywords, however, is that two people never chose the same keywords. The keywords then become useful only to people who already know the application well.

Practical keyword systems (such as that of VAX/NOTES for example) require keywords to be registered. This is already a step in the right direction. A linked system takes this to the next logical step. Keywords can be nodes which stand for a concept. A keyword node is then no different from any other node. One can link documents, etc., to keywords. One can then find keywords by finding any node to which they are related. In this way, documents on similar topics are indirectly linked, through their key concepts. A keyword search then becomes a search starting from a small number of named nodes, and finding nodes which are close to all of them.

It was for these reasons that I first made a small linked information system, not realising that a term had already been coined for the idea: "hypertext".

Personal Experience with Hypertext

In 1980, I wrote a program for keeping track of software with which I was involved in the PS control system. Called Enquire, it allowed one to store snippets of information, and to link related pieces together in any way. To find information, one progressed via the links from one sheet to another, rather like in the old computer game "adventure". I used this for my personal record of people and modules. It was similar to the application Hypercard produced more recently by Apple for the Macintosh. A difference was that Enquire, although lacking the fancy graphics, ran on a multiuser system, and allowed many people to access the same data.

Documentation of the RPC project

(concept)

Most of the documentation is available on VMS, with the two principle manuals being stored in the CERNDOC system.

- 1) includes: The VAX/NOTES conference VXCERN::RPC
- 2) includes: Test and Example suite
- 3) includes: RPC BUG LISTS
- 4) includes: RPC System: Implementation Guide
Information for maintenance, porting, etc.
- 5) includes: Suggested Development Strategy for RPC Applications
- 6) includes: "Notes on RPC", Draft 1, 20 feb 86
- 7) includes: "Notes on Proposed RPC Development" 18 Feb 86
- 8) includes: RPC User Manual
How to build and run a distributed system.
- 9) includes: Draft Specifications and Implementation Notes
- 10) includes: The RPC HELP facility
- 11) describes: THE REMOTE PROCEDURE CALL PROJECT in DD/OC

Help Display Select Back Quit Mark Goto_mark Link Add Edit

Fig 2. A screen in an Enquire scheme.

This example is basically a list, so the list of links is more important than the text on the node itself. Note that each link has a type ("includes" for example) and may also have comment associated with it. (The bottom line is a menu bar.)

Soon after my re-arrival at CERN in the DD division, I found that the environment was similar to that in PS, and I missed Enquire. I therefore produced a version for the VMS, and have used it to keep track of projects, people, groups, experiments, software modules and hardware devices with which I have worked. I have found it personally very useful. I have made no effort to make it suitable for general consumption, but have found that a few people have successfully used it to browse through the projects and find out all sorts of things of their own accord.

Hot spots

Meanwhile, several programs have been made exploring these ideas, both commercially and academically. Most of them use "hot spots" in documents, like icons, or highlighted phrases, as sensitive areas. touching a hot spot with a mouse brings up the relevant information, or expands the text on the screen to include it. Imagine, then, the references in this document, all being associated with the network address of the thing to which they referred, so that while reading this document you could skip to them with a click of the mouse.

"Hypertext" is a term coined in the 1950s by Ted Nelson [...], which has become popular for these systems, although it is used to embrace two different ideas. One idea (which is relevant to this problem) is the concept: "Hypertext": Human-readable information linked together in an unconstrained way.

The other idea, which is independent and largely a question of technology and time, is of multimedia documents which include graphics, speech and video. I will not discuss this latter aspect further here, although I will use the word "Hypermedia" to indicate that one is not bound to text.

It has been difficult to assess the effect of a large hypermedia system on an organisation, often because these systems never had seriously large-scale use. For this reason, we require large amounts of existing information should be accessible using any new information management system.

CERN Requirements

To be a practical system in the CERN environment, there are a number of clear practical requirements.

Remote access across networks.

CERN is distributed, and access from remote machines is essential.

Heterogeneity

Access is required to the same data from different types of system (VM/CMS, Macintosh, VAX/VMS, Unix)

Non-Centralisation

Information systems start small and grow. They also start isolated and then merge. A new system must allow existing systems to be linked together without requiring any central control or coordination.

Access to existing data

If we provide access to existing databases as though they were in hypertext form, the system will get off the ground quicker. This is discussed further below.

Private links

One must be able to add one's own private links to and from public information. One must also be able to annotate links, as well as nodes, privately.

Bells and Whistles

Storage of ASCII text, and display on 24x80 screens, is in the short term sufficient, and essential. Addition of graphics would be an optional extra with very much less penetration for the moment.

Data analysis

An intriguing possibility, given a large hypertext database with typed links, is that it allows some degree of automatic analysis. It is possible to search, for example, for anomalies such as undocumented software or divisions which contain no people. It is possible to generate lists of people or devices for other purposes, such as mailing lists of people to be informed of changes. It is also possible to look at the topology of an organisation or a project, and draw conclusions about how it should be managed, and how it could evolve. This is particularly useful when the database becomes very large, and groups of projects, for example, so interwoven as to make it difficult to see the wood for the trees.

In a complex place like CERN, it's not always obvious how to divide people into groups. Imagine making a large three-dimensional model, with people represented by little spheres, and strings between people who have something in common at work.

Now imagine picking up the structure and shaking it, until you make some sense of the tangle: perhaps, you see tightly knit groups in some places, and in some places weak areas of communication spanned by only a few people. Perhaps a linked information system will allow us to see the real structure of the organisation in which we work.

Live links

The data to which a link (or a hot spot) refers may be very static, or it may be temporary. In many cases at CERN information about the state of systems is changing all the time. Hypertext allows documents to be linked into "live" data so that every time the link is followed, the information is retrieved. If one sacrifices portability, it is possible so make following a link fire up a special application, so that diagnostic programs, for example, could be linked directly into the maintenance guide.

Non requirements

Discussions on Hypertext have sometimes tackled the problem of copyright enforcement and data security. These are of secondary importance at CERN, where information exchange is still more important than secrecy. Authorisation and accounting systems for hypertext could conceivably be designed which are very sophisticated, but they are not proposed here.

In cases where reference must be made to data which is in fact protected, existing file protection systems should be sufficient.

Specific Applications

The following are three examples of specific places in which the proposed system would be immediately useful. There are many others.

Development Project Documentation.

The Remote procedure Call project has a skeleton description using Enquire. Although limited, it is very useful for recording who did what, where they are, what documents exist, etc. Also, one can keep track of users, and can easily append any extra little bits of information which come to hand and have nowhere else to be put. Cross-links to other projects, and to databases which contain information on people and documents would be very useful, and save duplication of information.

Document retrieval.

The CERNDoc system provides the mechanics of storing and printing documents. A linked system would allow one to browse through concepts, documents, systems and authors, also allowing references between documents to be stored. (Once a document had been found, the existing machinery could be invoked to print it or display it).

The "Personal Skills Inventory".

Personal skills and experience are just the sort of thing which need hypertext flexibility. People can be linked to projects they have worked on, which in turn can be linked to particular machines, programming languages, etc.

Incentives and CALS

The State of the Art in Hypermedia

An increasing amount of work is being done into hypermedia research at universities and commercial research labs, and some commercial systems have resulted. There have been two conferences, Hypertext'87 and '88, and in Washington DC, the National Institute of Standards and Technology (NST) hosted a workshop on standardisation in hypertext, a follow up of which will occur during 1990.

The Communications of the ACM special issue on Hypertext contains many references to hypertext papers. A bibliography on hypertext is given in [NIST90], and a uucp newsgroup alt.hypertext exists. I do not, therefore, give a list here.

Browsing techniques

Much of the academic research is into the human interface side of browsing through a complex information space. Problems addressed are those of making navigation easy, and avoiding a feeling of being "lost in hyperspace". Whilst the results of the research are interesting, many users at CERN will be accessing the system using primitive terminals, and so advanced window styles are not so important for us now.

Interconnection or publication?

Most systems available today use a single database. This is accessed by many users by using a distributed file system. There are few products which take Ted Nelson's idea of a wide **docuverse** literally by allowing links between nodes in different databases. In order to do this, some standardisation would be necessary. However, at the standardisation workshop, the emphasis was on standardisation of the format for exchangeable media, nor for networking. This is prompted by the strong push toward publishing of hypermedia information, for example on optical disk. There seems to be a general consensus about the abstract data model which a hypertext system should use.

Many systems have been put together with little or no regard for portability, unfortunately. Some others, although published, are proprietary software which is not for external release. However, there are several interesting projects and more are appearing all the time. Digital's "Compound Document Architecture" (CDA), for example, is a data model which may be extendible into a hypermedia model, and there are rumours that this is a way Digital would like to go.

The US Department of Defence has given a big incentive to hypermedia research by, in effect, specifying hypermedia documentation for future procurement. This means that all manuals for parts for defence equipment must be provided in hypermedia form. The acronym CALS stands for **Computer-aided Acquisition and Logistic Support**.

There is also much support from the publishing industry, and from librarians whose job it is to organise information.

What will the system look like?

Let us see what components a hypertext system at CERN must have. The only way in which sufficient flexibility can be incorporated is to separate the information storage software from the information display software, with a well defined interface between them. Given the requirement for network access, it is natural to let this clean interface coincide with the physical division between the user and the remote database machine.

This division also is important in order to allow the heterogeneity which is required at CERN (and would be a boon for the world in general).

Accessing Existing Data

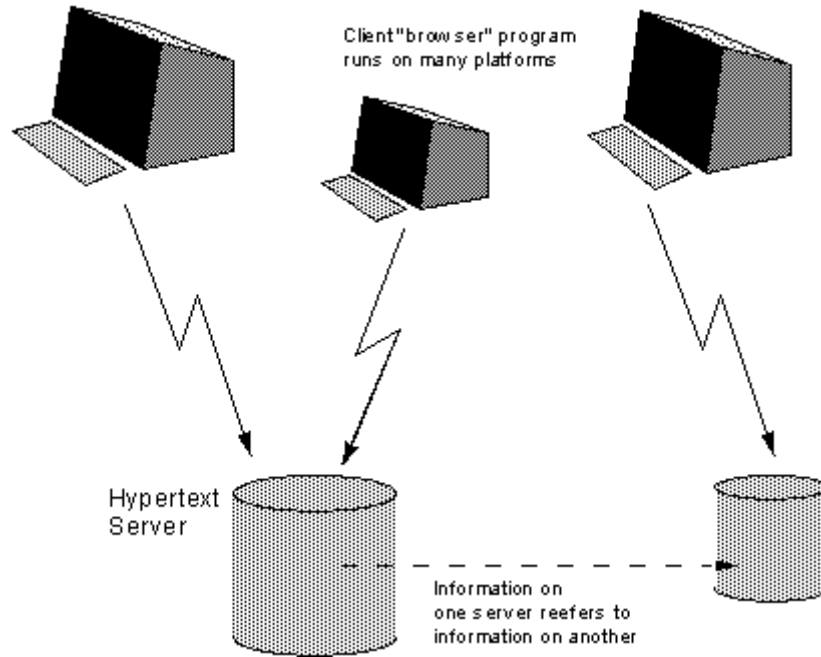


Fig 2. A client/server model for a distributed hypertext system.

Therefore, an important phase in the design of the system is to define this interface. After that, the development of various forms of display program and of database server can proceed in parallel. This will have been done well if many different information sources, past, present and future, can be mapped onto the definition, and if many different human interface programs can be written over the years to take advantage of new technology and standards.

The system must achieve a critical usefulness early on. Existing hypertext systems have had to justify themselves solely on new data. If, however, there was an existing base of data of personnel, for example, to which new data could be linked, the value of each new piece of data would be greater.

What is required is a gateway program which will map an existing structure onto the hypertext model, and allow limited (perhaps read-only) access to it. This takes the form of a hypertext server written to provide existing information in a form matching the standard interface. One would not imagine the server actually generating a hypertext database from an existing one: rather, it would generate a hypertext view of an existing database.

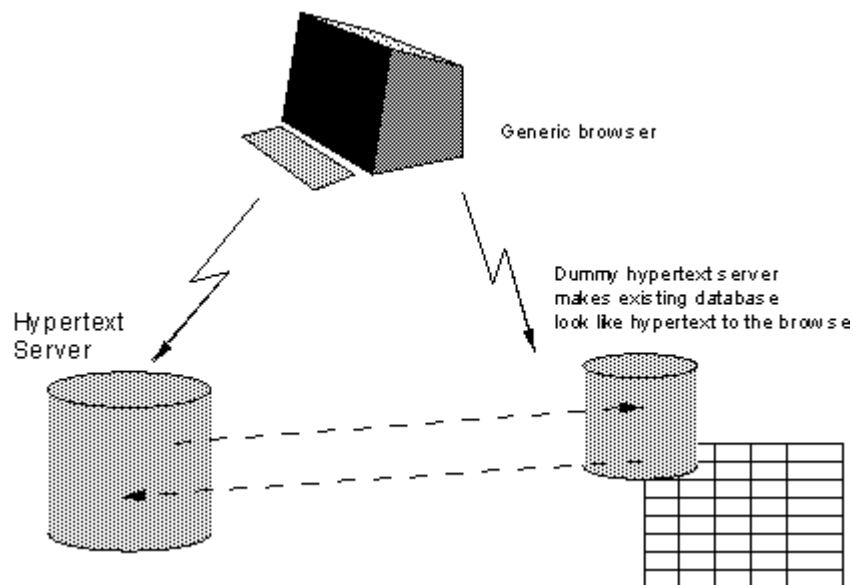


Fig 3. A hypertext gateway allows existing data to be seen in hypertext form by a hypertext browser.

Some examples of systems which could be connected in this way are

uucp News

This is a Unix electronic conferencing system. A server for uucp news could make links between notes on the same subject, as well as showing the structure of the conferences.

VAX/Notes

This is Digital's electronic conferencing system. It has a fairly wide following in FermiLab, but much less in CERN. The topology of a conference is quite restricting.

CERNDOC

This is a document registration and distribution system running on CERN's VM machine. As well as documents, categories and projects, keywords and authors lend themselves to representation as hypertext nodes.

File systems

This would allow any file to be linked to from other hypertext documents.

The Telephone Book

Even this could even be viewed as hypertext, with links between people and sections, sections and groups, people and floors of buildings, etc.

The unix manual

This is a large body of computer-readable text, currently organised in a flat way, but which also contains link information in a standard format ("See also..").

Databases

A generic tool could perhaps be made to allow any database which uses a commercial DBMS to be displayed as a hypertext view.

In some cases, writing these servers would mean unscrambling or obtaining details of the existing protocols and/or file formats. It may not be practical to provide the full functionality of the original system through hypertext. In general, it will be more important to allow read access to the general public: it may be that there is a limited number of people who are providing the information, and that they are content to use the existing facilities.

It is sometimes possible to enhance an existing storage system by coding hypertext information in, if one knows that a server will be generating a hypertext representation. In 'news' articles, for example, one could use (in the text) a standard format for a reference to another article. This would be picked out by the hypertext gateway and used to generate a link to that note. This sort of enhancement will allow greater integration between old and new systems.

There will always be a large number of information management systems - we get a lot of added usefulness from being able to cross-link them. However, we will lose out if we try to constrain them, as we will exclude systems and hamper the evolution of hypertext in general.

Conclusion

We should work toward a universal linked information system, in which generality and portability are more important than fancy graphics techniques and complex extra facilities.

The aim would be to allow a place to be found for any information or reference which one felt was important, and a way of finding it afterwards. The result should be sufficiently attractive to use that the information contained would grow past a critical threshold, so that the usefulness the scheme would in turn encourage its increased use.

The passing of this threshold accelerated by allowing large existing databases to be linked together and with new ones.

A Practical Project

Here I suggest the practical steps to go to in order to find a real solution at CERN. After a preliminary discussion of the requirements listed above, a survey of what is available from industry is obviously required. At this stage, we will be looking for a systems which are future-proof:

- portable, or supported on many platforms,
- Extendible to new data formats.

We may find that with a little adaptation, parts of the system we need can be combined from various sources: for example, a browser from one source with a database from another.

I imagine that two people for 6 to 12 months would be sufficient for this phase of the project.

A second phase would almost certainly involve some programming in order to set up a real system at CERN on many machines. An important part of this, discussed below, is the integration of a hypertext system with existing data, so as to provide a universal system, and to achieve critical usefulness at an early stage.

(... and yes, this would provide an excellent project with which to try our new object oriented programming techniques!) TBL March 1989, May 1990

References

[NEL67]

Nelson, T.H. "Getting it out of our system" in Information Retrieval: A Critical Review", G. Schechter, ed. Thomson Books, Washington D.C., 1967, 191-210

[SMISH88]

Smish, J.B and Weiss, S.F,"An Overview of Hypertext",in Communications of the ACM, July 1988 Vol 31, No. 7,and other articles in the same special "Hypertext" issue.

[CAMP88]

Campbell, B and Goodman, J,"HAM: a general purpose Hypertext Abstract Machine",in Communications of the ACM July 1988 Vol 31, No. 7

[ASKCYN88]

Akscyn, R.M, McCracken, D and Yoder E.A,"KMS: A distributed hypermedia system for managing knowledge in originations", in Communications of the ACM , July 1988 Vol 31, No. 7

[HYP88]

Hypertext on Hypertext, a hypertext version of the special Comms of the ACM edition, is available from the ACM for the Macintosh or PC.

[RN]

Under unix, type `man rn` to find out about the `rn` command which is used for reading uucp news.

[NOTES]

Under VMS, type `HELP NOTES` to find out about the VAX/NOTES system

[CERNDOC]

On CERNVM, type `FIND DOCFIND` for information about how to access the CERNDOC programs.

[NIST90]

J. Moline et. al. (ed.) Proceedings of the Hypertext Standardisation Workshop January 16-18, 1990, National Institute of Standards and Technology, pub. U.S. Dept. of Commerce

Appendix D: WorldWideWeb: Proposal for a HyperText Project

To:

P.G. Innocenti/ECP, G. Kellner/ECP, D.O. Williams/CN

Cc:

R. Brun/CN, K. Gieselmann/ECP, R. Jones/ECP, T. Osborne/CN, P. Palazzi/ECP, N. Pellow/CN, B. Pollermann/CN, E.M. Rimmer/ECP

From:

T. Berners-Lee/CN, R. Cailliau/ECP

Date:

12 November 1990

The attached document describes in more detail a Hypertext project.

HyperText is a way to link and access information of various kinds as a web of nodes in which the user can browse at will. It provides a single user-interface to large classes of information (reports, notes, data-bases, computer documentation and on-line help). We propose a simple scheme incorporating servers already available at CERN.

The project has two phases: firstly we make use of existing software and hardware as well as implementing simple browsers for the user's workstations, based on an analysis of the requirements for information access needs by experiments. Secondly, we extend the application area by also allowing the users to add new material.

Phase one should take 3 months with the full manpower complement, phase two a further 3 months, but this phase is more open-ended, and a review of needs and wishes will be incorporated into it.

The manpower required is 4 software engineers and a programmer, (one of which could be a Fellow). Each person works on a specific part (eg. specific platform support).

Each person will require a state-of-the-art workstation, but there must be one of each of the supported types. These will cost from 10 to 20k each, totalling 50k. In addition, we would like to use commercially available software as much as possible, and foresee an expense of 30k during development for one-user licences, visits to existing installations and consultancy.

We will assume that the project can rely on some computing support at no cost: development file space on existing development systems, installation and system manager support for daemon software.

T. Berners-Lee, R. Cailliau

WorldWideWeb: Proposal for a HyperText Project

T. Berners-Lee / CN, R. Cailliau / ECP

Abstract:

HyperText is a way to link and access information of various kinds as a web of nodes in which the user can browse at will. Potentially, HyperText provides a single user-interface to many large classes of stored information such as reports, notes, data-bases, computer documentation and on-line systems help. We propose the implementation of a simple scheme to incorporate several different servers of machine-stored information already available at CERN, including an analysis of the requirements for information access needs by experiments.

Introduction

The current incompatibilities of the platforms and tools make it impossible to access existing information through a common interface, leading to waste of time, frustration and obsolete answers to simple data lookup. There is a potential large benefit from the integration of a variety of systems in a way which allows a user to follow links pointing from one piece of information to another one. This forming of a web of information nodes rather than a hierarchical tree or an ordered list is the basic concept behind HyperText.

At CERN, a variety of data is already available: reports, experiment data, personnel data, electronic mail address lists, computer documentation, experiment documentation, and many other sets of data are spinning around on computer discs continuously. It is however impossible to "jump" from one set to another in an automatic way: once you found out that the name of Joe Bloggs is listed in an incomplete description of some on-line software, it is not straightforward to find his current electronic mail address. Usually, you will have to use a different lookup-method on a different computer with a different user interface. Once you have located information, it is hard to keep a link to it or to make a private note about it that you will later be able to find quickly.

Hypertext concepts

The principles of hypertext, and their applicability to the CERN environment, are discussed more fully in [1], a glossary of technical terms is given in [2]. Here we give a short presentation of hypertext.

A program which provides access to the hypertext world we call a browser. When starting a hypertext browser on your workstation, you will first be presented with a hypertext page which is personal to you : your personal notes, if you like. A hypertext page has pieces of text which refer to other texts. Such references are highlighted and can be selected with a mouse (on dumb terminals, they would appear in a numbered list and selection would be done by entering a number). When you select a reference, the browser presents you with the text which is referenced: you have made the browser follow a hypertext link :

(see Fig. 1: hypertext links).

That text itself has links to other texts and so on. In fig. 1, clicking on the GHI would take you to the minutes of that meeting. There you would get interested in the discussion of the UPS, and click on the highlighted word UPS to find out more about it.

The texts are linked together in a way that one can go from one concept to another to find the information one wants. The network of links is called a web . The web need not be hierarchical, and therefore it is not necessary to "climb up a tree" all the way again before you can go down to a different but related subject. The web is also not complete, since it is hard to imagine that all the possible links would be put in by authors. Yet a small number of links is usually sufficient for getting from anywhere to anywhere else in a small number of hops.

The texts are known as nodes. The process of proceeding from node to node is called navigation . Nodes do not need to be on the same machine: links may point across machine boundaries. Having a world wide web implies some solutions must be found for problems such as different access protocols and different node content formats. These issues are addressed by our proposal.

Nodes can in principle also contain non-text information such as diagrams, pictures, sound, animation etc. The term hypermedia is simply the expansion of the hypertext idea to these other media. Where facilities already exist, we aim to allow graphics interchange, but in this project, we concentrate on the universal readership for text, rather than on graphics.

Applications

The application of a universal hypertext system, once in place, will cover many areas such as document registration, on-line help, project documentation, news schemes and so on. It would be inappropriate for us (rather than those responsible) to suggest specific areas, but experiment online help, accelerator online help, assistance for computer center operators, and the dissemination of information by central services such as the user office and CN and ECP divisions are obvious candidates. WorldWideWeb (or W3) intends to cater for these services across the HEP community.

Scope: Objectives and non-Objectives

The project will operate in a certain well-defined subset of the subject area often associated with the "Hypertext" tag. It will aim:

- to provide a common (simple) protocol for requesting human readable information stored at a remote system, using networks;
- to provide a protocol within which information can automatically be exchanged in a format common to the supplier and the consumer;
- to provide some method of reading at least text (if not graphics) using a large proportion of the computer screens in use at CERN;
- to provide and maintain at least one collection of documents, into which users may (but are not bound to) put their documents. This collection will include much existing data. (This is partly to give us first hand experience of use of the system, and partly because members of the project will already have documentation for which they are responsible)
- to provide a keyword search option, in addition to navigation by following references, using any new or existing indexes (such as the CERNVM FIND indexes). The result of a keyword search is simply a hypertext document consisting of a list of references to nodes which match the keywords. to allow private individually managed collections of documents to be linked to those in other collections. to use public domain software wherever possible, or interface to proprietary systems which already exist.
- to provide the software for the above free of charge to anyone.

The project will not aim

- to provide conversions where they do not exist between the many document storage formats at CERN, although providing a framework into which such conversion utilities can fit;
- to force users to use any particular word processor, or mark-up format;
- to do research into fancy multimedia facilities such as sound and video;
- to use sophisticated network authorisation systems. data will be either readable by the world (literally), or will be readable only on one file system, in which case the file system's protection system will be used for privacy. All network traffic will be public.

Requirements Analysis

In order to ensure response to real needs, a requirements analysis for the information access needs of a large CERN experiment will be conducted at the very start, in parallel with the first project phase.

This analysis will at first be limited to the activities of the members of the Aleph experiment, and later be extended to at least one other experiment. An overview will be made of the information generation, storage and retrieval, independent of the form (machine, paper) and independent of the finality (experiment, administration).

The result should be:

- lists of sources, depots and sinks of information,
- lists of formats,
- diagrams of flow,
- statistics on traffic,
- estimated levels of importance of flows,
- lists of client desires and / or suggested improvements,
- estimated levels of satisfaction with platforms,
- estimated urgency for improvements.

This analysis will itself not propose solutions or improvements, but its results will guide the project.

Architecture

The architecture of the hypertext world is one of data stored on server machines, and client processes on the same or other machines. The machines are linked by some network (fig. 2). Fig. 2: proposed model for the hypertext world A workstation is either an independent machine in your office or a terminal connected to a close-by computer, and connected to the same network. The servers are active processes that reply to requests. The hypertext data is explicitly accessible to them. Servers can be many on the same computer system, but then each caters to a specific hypertext base. Clients are browser processes, usually but not necessarily on a different computer system. Information passed is of two kinds: nodes and links.

Building blocks

Browsers and servers are the two building blocks to be provided.

A browser is a native application program running on the client machine:-

- it performs the display of a hypertext node using the client hardware and software environment. For example, a Macintosh browser will use the Macintosh interface look-and-feel.
- it performs the traversal of links. For example, when using a Macintosh to browse on CERNVM FIND it will be the Macintosh browser which remembers which links were traversed, how to go back etc., whereas the CERNVM server just responds by handing the browser nodes, and has no idea of which nodes the user has visited.
- it performs the negotiation of formats in dialog with the server. For example, a browser for a VT100 type display will always negotiate ASCII text only, whereas a Macintosh browser might be constructed to accept PostScript or SGML.

A server is a native application program running on the server machine:-

- it manages a web of nodes on that machine.-
- it negotiates the presentation format with the browser, performing on-the-fly (or cached) conversions from its own internal format, if any..

Operation

A link is specified as an ASCII string from which the browser can deduce a suitable method of contacting an appropriate server. When a link is followed, the browser addresses the request for the node to the server. The server therefore has nothing to know about other servers or other webs and can be kept simple.

Once the server has located the requested node, it will know from the node contents what the node's format is (eg. pure ASCII, marked-up, word processor storage and which word processor etc.). The server then begins a negotiation with the browser, in which they decide between them what format is acceptable for display on the user's screen. This negotiation will be based only on existing conversion programs and formats: it is not in the scope of W3 to write new converters. The last resort in the negotiation is the binary transfer of the node contents to a file in the user's file space. Negotiating the format for presentation is particular to W3.

Project phases

Provided with resources mentioned below, we foresee the first two phases of the project as achieving the following goals:

Phase 1 -- Target: 3 months from start

- Browsers on dumb terminal to open readership to anyone with a computer or PC.(?)
- Browsers on vt220 terminals to give cursor-oriented readership to a very large proportion of readers; A browser on the Macintosh in the Macintosh style; A browser on the NeXT using the NeXTStep tools, as a fast prototype for ideas in human interface design and navigation techniques.
- A server providing access to the world of Usenet/Internet news articles. *
- A server providing access to all the information currently stored on CERNVM and mentioned in the FIND index. This should include CERN program library notes, IBM and CERN CMS help screens, CERN/CN writeups, Computer Newsletter articles, etc.
- A server which may be installed on any machine to allow files on that machine to be accessed as hypertext.
- The ability for users to write, using markup tags, their own hypertext for help files. No other hypertext editing capability will necessarily be implemented in this phase.
- A gateway process to allow access between the Internet and DECnet protocol worlds.
- A set of guidelines on how to manage a hypertext server.
- A requirements analysis of the information access needs for a large experiment.

At this stage, readership is universal, but the creation of new material relies on existing systems. For example, the introduction of new material for the FIND index, or the posting of news articles will use the same procedures as at present. we gain useful experience in the representation of existing data in hypertext form, and in the types of navigational and other aids appreciated by users in high energy physics.

Phase 2 -- Target: 6 months from start

In this important phase, we aim to allow

- The creation of new links and new material by readers. At this stage, authorship becomes universal.
- A full-screen browser on VM/XA for those using CERNVM, and other HEP VM sites;
- An X-window browser/editor, giving the sophisticated facilities originally prototyped under NeXTStep to the wide X-based community. (We imagine using OSF/Motif subject to availability)
- The automatic notification of a reader when new material of interest to him/her has become available. This is essential for news articles, but is very useful for any other material.

The ability of readers to create links allows annotation by users of existing data, allows to add themselves and their documents to lists (mailing lists, indexes, etc). It should be possible for users to link public documents to (for example) bug reports, bug fixes, and other documents which the authors themselves might never have realised existed. This phase allows collaborative authorship. It provides a place to put any piece of information such that it can later be found. Making it easy to change the web is thus the key to avoiding obsolete information. One should be able to trace the source of information, to circumvent and then to repair flaws in the web.

Resources required

1. People

The following functions are identifiable. They do not necessarily correspond to individuals on a one to one basis. The initials in brackets indicate people who have already expressed an interest in the project and who have the necessary skills but do not indicate any commitment as yet on their part or the part of their managers. We are of course very open to involvement from others.

- System architect. Coordinate development, protocol definition, etc; ensures integrity of design. (50% TBL?) Market research and product planner. Discuss the project and its features with potential and actual users in all divisions. Prepare criteria for feature selection and development priority. (50% RC?)
- Hyper-Librarian. Oversees the web of available data, ensuring its coherency. Interface with users, train users. Manages indexes and keyword systems. Manages data provided by the project itself. (100% KG?)
- Software engineer: NeXTStep. Provide browser/editor interface under the NeXTStep human interface tools. Experiment with navigational aids. Keep a running knowledge of the NeXTStep world. (50% TBL?)
- Software engineer: X-windows and human interface. Provide browser/editor human interface under OSF/Motif. Respond to user suggestion for ease of use improvements and options. Create an aesthetic, practical human interface. Keep a running knowledge of the X world. (75%RJ?)
- Software engineer: IBM mainframe. Provide browser service on CERNVM and other HEP VM sites. Maintain the FIND server software. Keep up a running knowledge of the CMS, Rexx world. (75% BP?)
- Software engineer: Macintosh. Provide browser/editor for the mac, using whatever tools are appropriate (Thnk-C, HyperCard, etc?). (50%RC?)
- Software engineer: C. Help write code for dumb terminal or vt100 browsers, and portable browser code to be shared between browsers. This could include a technical student project. (100% NP? + A.N.Other?)

We foresee that a demand may arise for browsers on specific systems, for specific customizations, and for servers to make specific existing data available online as hypertext. We intend to enthusiastically support such widening of the web. Of course, we may have to draw on more manpower and specific expertise in these cases.

2. Other resources

We will require the following support in the way of equipment and services.

- We feel it is important for those involved in the project to be able to work close to each other and exchange ideas and problems as they work. An office area or close group of offices is therefore required.
- Each person working on the project will require a state-of-the-art workstation. Experience shows that a workstation has to be upgraded in some way every two years or so as software becomes more cumbersome, and memory/speed requirements increase. This, and the cost of software upgrades, we foresee as reasonable expenses. We imagine using a variety of types of workstation as we provide software on a variety of machines, but otherwise NeXTs. For VMS machines, we would like the support of an existing VAXcluster to minimize our own system management overheads.
- We would like to be able to purchase licenses for commercial hypertext software where we feel this could be incorporated into the project, and save development and maintenance time, or where we feel we could gain useful experience from its use. (Approximate examples are: Guide license: CHF750; KMS full author license CHF1500, evaluation kit CHF100. FrameMaker: CHF2000)
- We will require computing support. In particular, we will require a reliable backed up NFS (or equivalent) file server support for our development environment. We will also need to run daemon software on machines with Internet, DECnet and BITNET connectivity, which will require a certain amount of support from operators and system managers.

Future paths

- The two phases above will provide an extremely useful set of tools. Though the results seem ambitious, the individual steps necessary are well within our abilities with available technology. Future developments which would further enhance the project could include:
- Daemon programs which run overnight and build indexes of available information.
- A server automatically providing a hypertext view of a (for example Oracle) database, from a description of the database and a description (for example in SQL) of the view required.
- Work on efficient networking over wide areas, negotiation with other sites to provide compatible online information.
- A serious study of the use and abuse of the system, the sociology of its use at CERN.

References

- [1] T. Berners-Lee/CN, HyperText and CERN . An explanation of hypertext, and why it is important for CERN. A background document explaining the ideas behind this project.
- [2] T. Berners-Lee/CN, Hypertext Design Issues . A detailed look at hypertext models and facilities, with a discussion of choices to be made in choosing or implementing a system.
- [3] Other documentation on the project is stored in hypertext form and which leads to further references.

