Vague but exciting ...

CERN DD/OC　　Tim Berners-Lee, CERN/DD

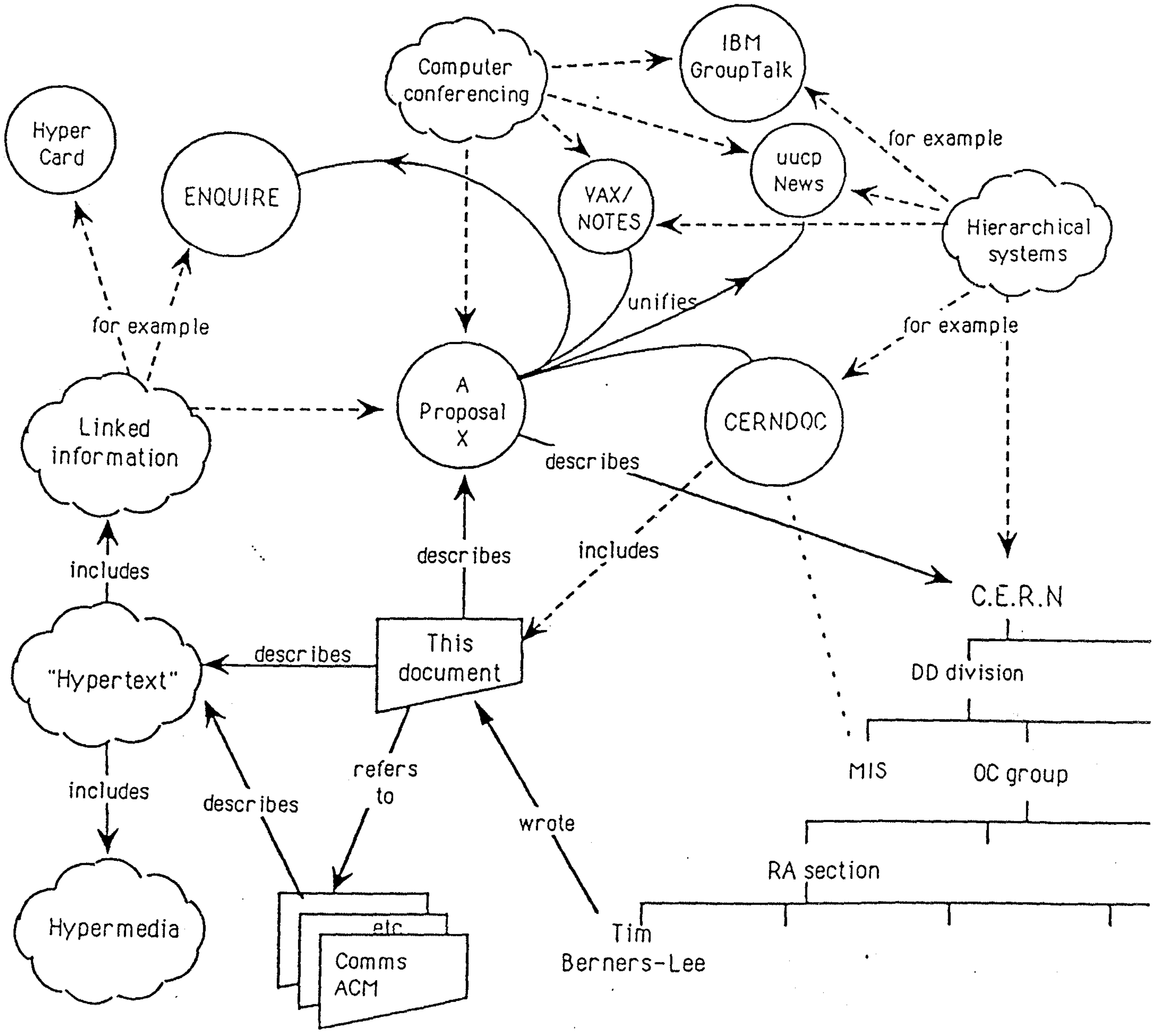
Information Management: A Proposal　　March 1989

Information Management: A Proposal

Abstract

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 sytstem.

Keywords: Hypertext, Computer conferencing, Document retrieval, Information management, Project control



I like the browsing style this should make possible. Intuitively right and potentially user-friendly. But it could be frustrating - how do we know we can get it right? Demos of existing systems? Invaluable tool for services

1. 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 unforseen 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 accute 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[1].

[1] The same has been true, for example, of electronic mail gateways, document preparation, and heterogeneous distruibuted programming systems.

2. 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.

Yes (but)**

We can call the circles nodes, and the arrows links. Suppse 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 decribes 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,[2] have different significance in various sorts of conventional diagrams

* Formal documentation needed
Private / institutional systems
How to get stuff put in ?

** But it complicates life for the busy user
(a classic tradeoff ...)

[2] Linked information systems have entities and relationships. There are, however, many differences between such a system and an "Entity Relationship" database system. For one thing, the information stored in a linked system is largely comment for human readers. For another, nodes do not have strict types which define exactly what relationships they may have. Nodes of similar type do not all have to be stored in the same place.

Diagram	Nodes are...	Arrows mean...
Family Tree	People	"Is parent of"
SASD diagram	Software module	"Passes data to"
Dependency chart	Software modules	"Depends on"
Organisation 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. Yes

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.

2.1 The problem with trees Yes

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_LINE OPTIONS DEFAULTS PRAGMAS"

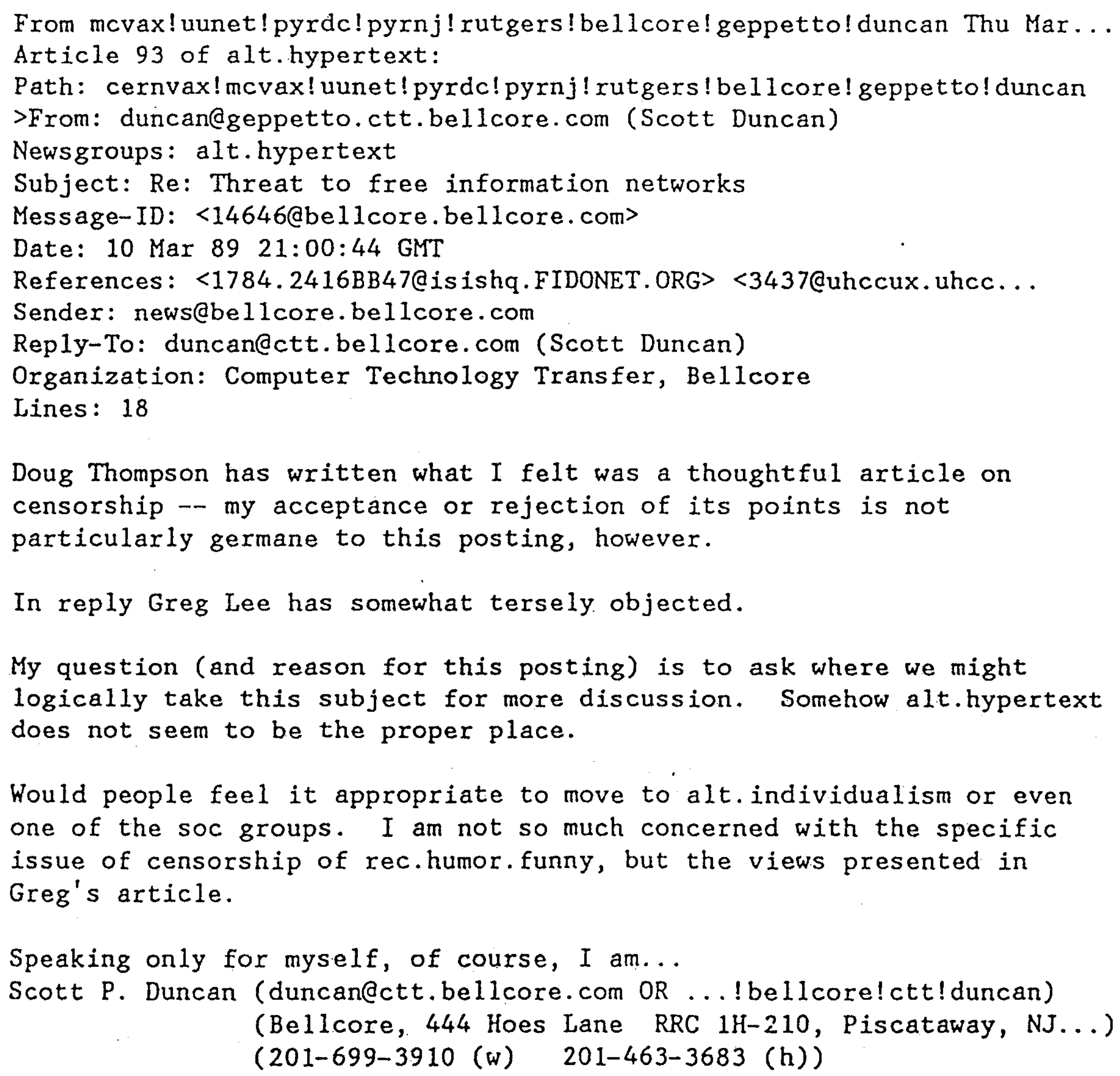
and it is necessary to leave the system and reenter it. What was needed was a link from one node to another, because in this case the information was not naturally organised into a tree.

Another example of a tree-structured system is the uucp News system ('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 Figure 1)

2.2 The problem with keywords Yes

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.



```
From mcvax!uunet!pyrdc!pyrnj!rutgers!bellcore!geppetto!duncan Thu Mar...
Article 93 of alt.hypertext:
Path: cernvax!mcvax!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))
```

Figure 1: A note 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 is expresses a problem with the strict tree structure of the scheme: this discussion is related to several areas. Note the "References", "From" and "Subject" fields can all be used to generate links.

Nice idea

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.

(but proliferation problem is still there)

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 technique: "Hypertext".

3. 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.

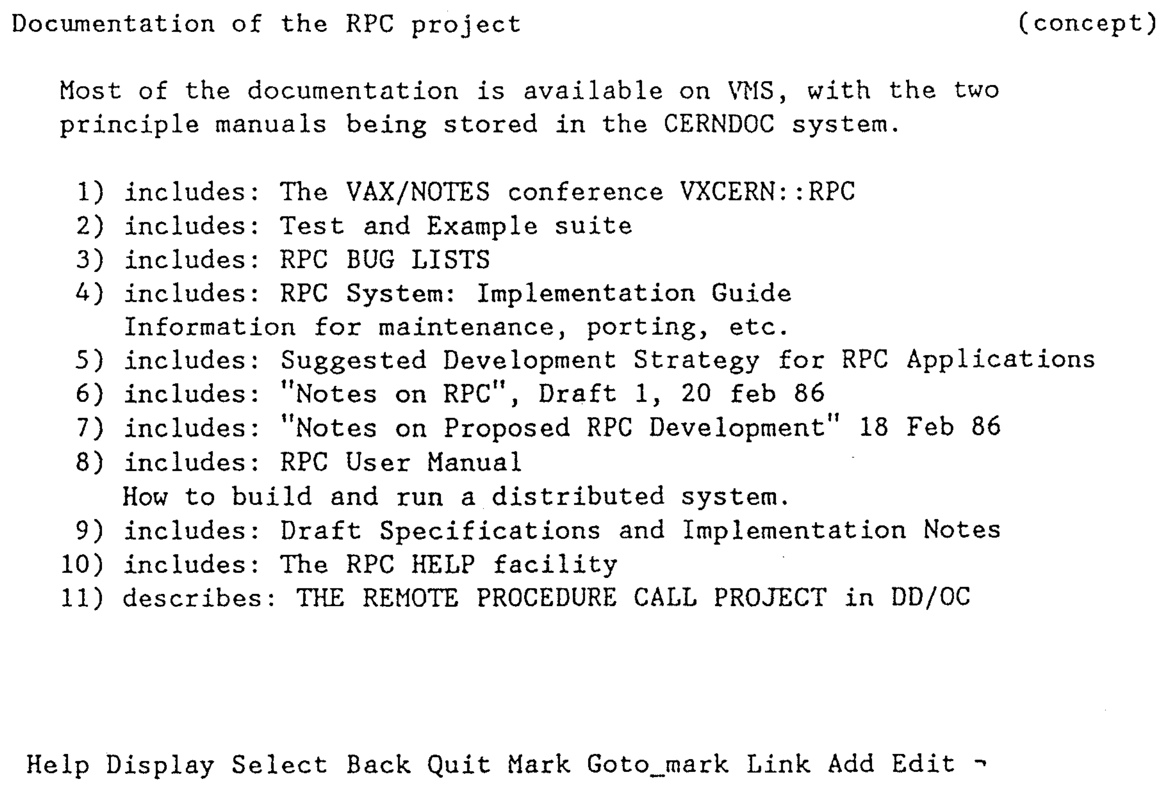
Soon after my re-arrival at CERN in the DD division, I found that the environment was similar to that in PS, and I missed *Enquire*. I therefore produced a version for the VMS, and have used it to keep track of projects, people, groups, experiments, software modules and hardware devices with which I 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.

Meanwhile, several programs [...] have been made exploring these ideas, both commercially and academically. Many of these have concentrated largely on the human interface aspects, and the methods of presenting linked information to a person with a workstation. *

"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 of information being linked together in an unconstrained way. The other idea, which is less immediately relevant and largely a question of technology and time, is of multimedia documents, including graphics, speech and video. I will not discuss this latter aspect ("Hypermedia") futher here.

It has been difficult to assess the effect of a large 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. ✓

* This is important — can we pick anything up?



```
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 ¬

Figure 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 itelf. 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.)

4. Requirements

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

1. *REMOTE ACCESS ACROSS NETWORKS.*

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

2. *HETEROGENEITY*

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

This is like e-mail, but who will implement-
and port all the user agents ?

3. *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.

4. *ACCESS TO EXISTING DATA*

If we provide acess to existing databases as though they were in hypertext form, the system will get off the ground quicker. This is discussed futher in an appendix.

5. *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.

6. *BELLS AND WHISTLES* I'm not convinced.

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

4.1 Client/Server Model

I don't understand what this interface is.

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.

(A client/server split at this level also makes multi-access more easy, in that a single server process can service many clients, avoiding the problems of simultaneous access to one database by many different users.)

Well, er, yes ...

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 procede 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.

If, in the future, this work is repeated with the benefit of hindsight and experience (and international cooperation?), it may be done differently. However, one would imagine that the gateway technique would allow the new interface standard to be introduced painlessly.

Important aspects of the standard interface are

- That it should be a superset of most existing and seriously concievable information systems; WOW !
- That advanced features should be mappable in a defined way onto a simple subset of features; ?
- That it should be open to extension;

- That it should be open in that it does not impose arbitrary constraints on any associated software apart from its own purpose. It should make no reference to particular properties of operating systems, etc.

- It should use existing standards whever possible for document and graphics representation, etc.

4.2 Data analysis

Fun but a side-issue

An intruiging 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 representaced 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 organistaion in which we work.

4.3 Non requirements

Right

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.

5. Summary

This proposal describes a universal linked information system, in which generality and portability are more important than fancy graphics techniques and complex extra facilities.

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

but maybe we could assume X-Window ?

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

And now ?

Project - manpower - timescales ?
Survey phase ?
Tech. specs.

Institutional framework ?
Service providers list
UO ? AS ? MIS ... danger. Frigo ?
Non-CERN possibilities ?

Structure ?

Academic side ?

Identify existing data to start with ? Group scheme ?

How does this compare with conventional DBMS or data modelling ? Queries allow a sort of browsing after all but with discipline imposed on the database itself. Is this discipline useful or constraining ? Will this scheme degenerate into unmanageable spaghetti ? What can be done to prevent / rectify this ? Will there be performance problems ?

What about computerised teaching schemes ? Can we learn from them ? Pick anything up ?

Appendix A

Accessing Existing Data

Absolutely right.

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 personel, 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 and existing one: rather, it would generate a hyperytext view of an existing database.

Oracle on Mac ?

Some examples are

1. *UUCP NEWS*

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

2. *VAX/NOTES* graphics again.

   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.

3. *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.

4. *FILE SYSTEMS*

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

5. *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.

6. *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..").

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.

* Problem of documents sharing material is also soluble.
How does text processing fit in ?

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. Nice

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. Howver, we will lose out if we try to constrain them, as we will exclude systems and hamper the evolution of hypertext in general.

OC + UO databases

VMS doc on CD-rom } doesn't something exist?
VMS HELP

A thought – structure of a document.

Appendix B

Specific Applications

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

- *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, sytems 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.

Clearing-house

Appendix C

Project ingredients: old and new

Many parts of the proposed system should be available from existing sources. A search of related software may well extend this list. In some cases (marked * below) the technology is new, and although it is servicable it may have to be replaced when something better comes along. In others, acceptable standards or software probably exists (marked **)

The work required is broken into parts as discussed above: the interface specification, and initial pilot client and server programs.

C.1 Interface specification

1. *REFERENCE LOGICAL DATA MODEL DESIGN* *

   a. Single/bidirectional links. There are advantages if a link in one direction automatically is accessible in the other direction. This often doubled the usefulness of the original link. However, it causes problems if the make of the link only has write access to one of the things he is linking.

   b. Private and public links. It is necessary to be able to make one's own private links between public objects. These links may be stored locally on one's own machine, for efficiency and privacy. Their implemenattion must be made consistent with public links stoed in the servers.

   c. Overlaying of equivalent nodes in different databases. As databases growm it often comes about that the same thing crops up in both. After a certain time, this becomes aparent, and the two should be merged. However, it is still useful (for efficiency, and protection reasons) to keep entried in both databases. Therefore, one requires a "virtual merging" of nodes, so that the information about the same thing is displayed together, subject to a particular client's rights and ability to access the various schemes.

What ?

2. Choice of document representation formats

   a. Format negotiation, coersion to lowest common format. A basic commom standard must exist which can be displayed on a 24x80 character screen using ASCII characters. All servers must be prepared to produce data in that format. Other formats may be included, such as marked up text and graphics, subject to agreement between the client and server. In this way, new formats can be introduced as they become available.

   b. Reference to existing standards **

Content v. structure.

3. *LINK DEFINITION FORMAT*

   a. Network address of server*

   b. Node reference within server

   c. Link identification within node

    d. Logical naming for network independence

4. *COMMUNICATION STANDARDS*

    a. Heterogeneous RPC **

    b. Network naming standards *

5. Human Interface Window systems **, etc.

C.2 Template client

1. Human Interface tools (X – windows etc) **

2. Text/Graphics editors **

3. Stashing techniques *

C.3 Template server

1. Database technology **

C.4 Template server gateway

1. Require access to existing protocols **

References

1. 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

2. 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.

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

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

```
          - Hypertext on Hypertext
        - existing systems - rn, NOTES, etc, CERNDOC
```