Supporting Collaboration through Multimedia Digital Document Archives

Brian R Gaines

Knowledge Science Institute, University of Calgary

Version 1.0, November 1994

Supporting Collaboration through Multimedia Digital Document Archives

Brian R Gaines

Knowledge Science Institute, University of Calgary
Calgary, Alberta, Canada T2N 1N4
gaines@cpsc.ucalgary.ca

Abstract: Increasing ease of access to the Internet is making it feasible for geographically dispersed communities to work closely together, coordinating their activities through electronic mail, digital document archives, and access to remote computing facilities. The support of the collaboration through digital discourse also makes it feasible to disseminate the results of the collaboration to others through the same media, for example through World Wide Web or CD-ROM. This presentation reports on practical experience of supporting a number of communities and projects through the use of the Internet and CD-ROM. It gives an overview of the technologies available, their accessibility, ease of use and impact on collaborative activities. It focuses on the practical problems that arise, the limitations of existing technologies, and how these may be overcome. Examples are given of projects encompassing a range of multimedia digital technologies from list servers, through World Wide Web document archives, to production in a few days of CD-ROMs containing movies and digitized foils and documents giving a complete account of working meetings.

1 INTRODUCTION

The Knowledge Science Institute (KSI) was set up in 1985 with a dual mandate (Gaines, 1985): to study and model the *knowledge economy* (Bell, 1973; Machlup, 1980) with a view to forecasting trends and modeling knowledge processes; and to innovate in selected areas that might lead to insights into the impact of *knowledge support systems* (Gaines, 1990). One conclusion of the KSI studies is that a major anachronism in human knowledge processes has, until recently, been the continued reliance on paper media for knowledge dissemination (Gaines, 1993). While book production technology has gone through several major technological advances, the end-product is basically the same as it was for Gutenberg and Caxton over five hundred years ago. Similarly, scholarly journals have changed little in their presentation and operation since the inception of the *Philosophical Transactions* of the Royal Society in 1665.

When one considers the limitations of the paper medium in communicating knowledge, and contrasts this with the capabilities of current multimedia computer and communication technology, the incentives to make major changes in scholarly publishing are very high. Books and journals primarily support typographic text and monochrome diagrams and photographs, and require separate indexing for purposes of contextual search. Computer-readable multimedia documents support the same quality of typographic text, color diagrams and photographs, sound and video, animation, simulation, computation, and are intrinsically contextually searchable. They can be produced and distributed on CD-ROM at a substantially lower costs than paper books or journals, and can be read on personal computers and selectively printed on laser printers. They can also be produced and distributed on the Internet, providing an active medium supporting many forms of scholarly discourse.

The objective of the KSI research program reported in this article is to provide a next generation of open architecture knowledge support systems (Gaines, 1994a) supporting collaboration in scholarly communities through information technology with a view to achieving systematic acceleration of human knowledge processes. Specifically, a new layer of knowledge level tools has been developed that complements existing list servers and electronic document archives with research coordination facilities. The tools are designed to support the knowledge processes of scientific communities through active document, hypermedia, concept mapping, knowledge representation and active agent technologies. They use existing Internet protocols for mail, file transfer and hypermedia to integrate with related systems. They provide import and export facilities for common document formats to integrate with existing document production. The data formats are designed for parallel use on CD-ROM to provide alternative means of distribution to the Internet.

The pragmatic objective is to achieve widespread use of greatly improved facilities for scientific discourse and collaborative project management. A deeper objective is to study the impact of the proposed technologies on actual research practice with a view to designing a next generation of tools targeted on the changes in patterns of research behavior already apparent as new information technologies have come to play a significant role in scholarship (Manitoba, 1993; Renear and Bilder, 1993).

Figure 1.1 shows the general situation and support technologies studied in this article. A goal-directed, geographically dispersed community is engaged in a purposeful activity involving *communicative action* (Habermas, 1981)—that is, discourse in which mutual understanding is a joint objective. The *process* of discourse in a collaborative community can be supported through technologies such as electronic mail (email) and mailing list servers. The *product* of this discourse is captured in multimedia materials and document archives whose dissemination can be supported through technologies such as CD-ROMs and the Internet file transfer protocol (FTP). Hybrid client-server technologies such as those of the World Wide Web may be used to support both process and product through structured discourse with archives indexed and linked to a record of the discourse.

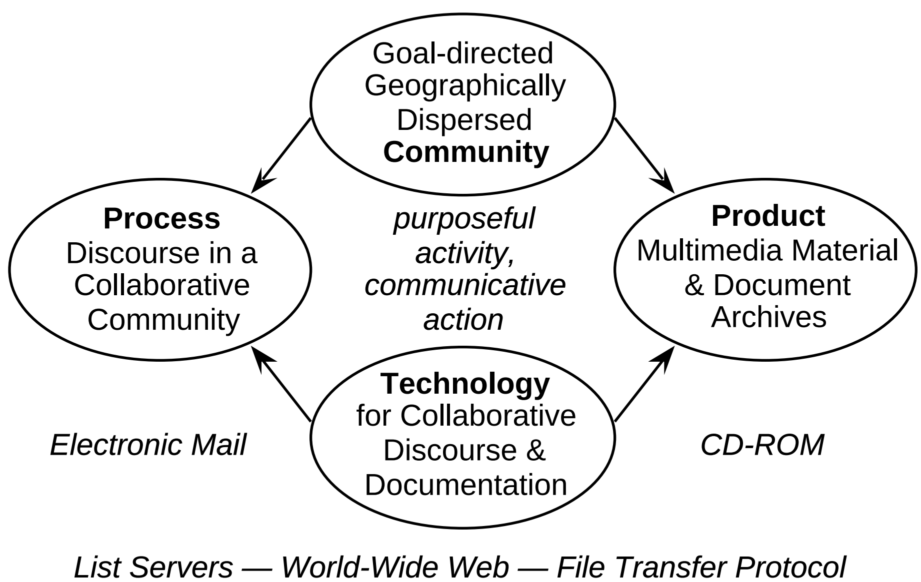


Figure 1.1 Supporting the knowledge processes of a dispersed community

The principles and applications of the knowledge-level tools developed by the KSI have been reported in a number of articles which are available in publications (Gaines and Shaw, 1993; Gaines and Shaw, 1994c; Gaines and Shaw, 1994b; Kremer and Gaines, 1994; Shaw and Gaines, 1994) and through FTP (ftp://ksi.cpsc.ucalgary.ca/KSI/). This article focuses on practice and experience with commonly available technologies and tools. Its objective is to help others produce multimedia presentations in QuickTime, on CD-ROM, and through World Wide Web.

2 BACKGROUND TO STUDIES—IMS GNOSIS TEST CASE

To provide a context for the projects used as examples in this article, it is relevant to give a brief overview of the international GNOSIS community whose support has been a focus of the major projects. IMS TC7 'GNOSIS', is one of 6 one-year test cases under the international Intelligent Manufacturing Systems research program which started in the second quarter of 1993. The project (GNOSIS, 1994) involves over 100 participants in 31 industry and university organizations in 14 countries, with the objective of developing a *post mass production manufacturing paradigm* involving *reconfigurable artifacts*. The project has made extensive use of electronic mail and electronic document archives to coordinate its activities, and the studies reported are part of an investigation to improve such coordination in the main 10-year study commencing in 1995.

Figure 2.1 shows the partners involved in the GNOSIS consortium together with their geographic locations. Figure 2.2 shows the meetings during the first year of the project and the cost impact of coordinating such a dispersed project—that meetings accounted for some $1.1M of the estimated $5M budget for the 1-year test case. The meetings and associated costs were probably necessary in the initial stages of a major collaborative venture involving partners many of whom had never previously interacted. However, the costs involved draw attention to two considerations relevant to multi-media communication:

- Each major meeting should be treated as an expensive resource and every effort made to maximize the value received. In particular, the discourse at meetings should be recorded, indexed and archived, as should the slides presented, documents circulated, and so on.
- Other communication media such as telephone, tele- and video-conferencing, fax, email, list servers, document archives, and so on, should be used effectively to reduce the need for physical meetings.

Figure 2.3 shows the digital document archives and list servers set up to support the GNOSIS community. The logic behind the locations is interesting. Technically, the primary concern is to have a site with a reasonably high-speed connection to the Internet and spare capacity to carry additional traffic. There may also be considerations relating to specialist resources—for example, some GNOSIS partners in Europe did not have email access, and the Fraunhofer Institute was able to offer an automatic email-to-fax connection that was most cost-effectively located in Europe. Administratively, an archive or list server can be managed over the Internet from anywhere in the world. However, a local administrator may have more influence in ensuring that the facilities are restored with high priority in the even of a system fault, such as computer or disk drive failure. Socially, it may be desirable that the sites are distributed so that there is no appearance of control of a major resource by any particular special-interest group.

Region: Japan
Mitsubishi Electric Corp.
Nissan Motors Ltd
Fuji Xerox Ltd
Kajima Corp.
Shimizu Corp.
The University of Tokyo
Tokyo Institute of Technology
Kyushu Institute of Technology

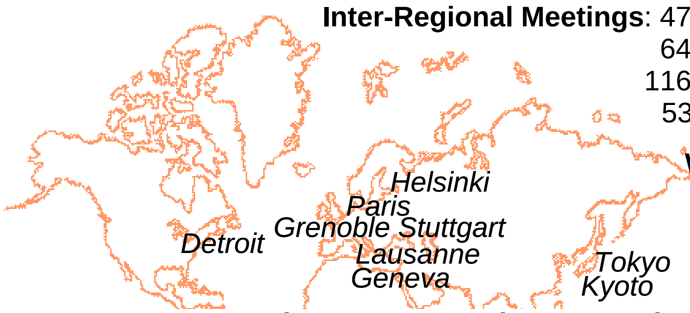
Region: EFTA
ABB, Finland
Tehdasmallit Oy, Finland
Synergy Center Ltd, Finland
SIMSON, Finland
VTT, Finland
Tampere University, Finland
EPFL, Switzerland
ETHZ, Switzerland

Region: EC
ADEPA, France
Télémécanique, France
ARM Conseil, France
IBM, France
ITMI, France
LLP/CESALP, France
Fraunhofer Inst. IPA, Germany
MAN Roland, Germany
Brose, Germany
BICC plc, UK
Cambridge University, UK

Region: Canada
Alberta Research Council
DME, University of Calgary
KSI, University of Calgary

Region: USA
Deneb Robotics Inc

Figure 2.1 The GNOSIS consortium



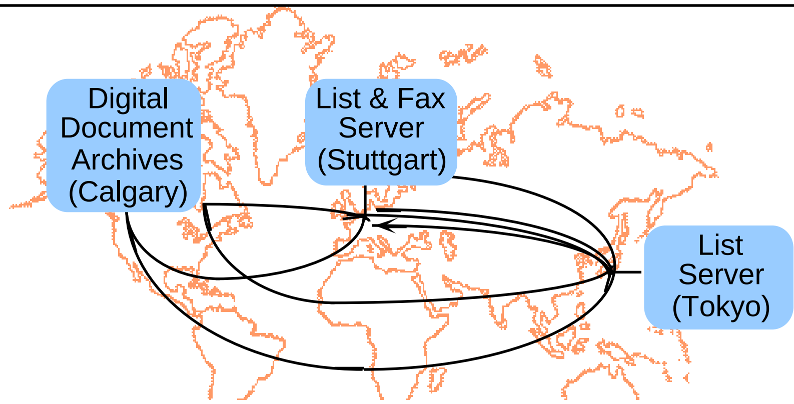
Inter-Regional Meetings: 47 at Stuttgart, February 11-13,
64 at Helsinki, August 2-4,
116 at Kyoto, November 8-12,
53 at Grenoble, Jan 31-Feb 4, 1994

Workshops:
TW1: 20 at Detroit, April 27-28; 20 at Geneva, September 27-28
TW2: 17 at Paris, April 15-16; 18 at Lausanne, June 10-11
TW3: 14 at Detroit, June 1-2
TW4: 24 at Tokyo, March 20-21; 19 at Tokyo, June 30-July 1
TW5: 12 at Stuttgart, May 11-12

MEETING COSTS	People	Person Days	Salary Cost US$K	Travel Cost US$K
In Organization	27	54	27	-
In Country	73	219	109	36
On-Content	122	488	244	122
Trans-Continent	112	560	280	336
TOTALS	1321	660	494	

Estimated Total Meeting Cost US$1,154,000

Figure 2.2 GNOSIS meeting costs in first year of operation



Partners send email to group on list servers located at two partner sites.

List server forwards mail to all partners in group by email or fax.

Lag between sending and receiving is less than 2 minutes.

List servers also archive all mail and make it accessible through email requests to registered partners at any time.

Figure 2.3 GNOSIS digital archives and list servers

3 DIGITAL MOVIES AS TUTORIAL DEMONSTRATIONS

One of the common activities at the initial GNOSIS meetings was for partners to give demonstrations to one another of ongoing research in their laboratories. The typical form of demonstration is a presentation of some system in operation together with a commentary on what is happening, why, and how this illustrates the overall objectives and state of the research. This type of tutorial demonstration is ideally captured as a movie which can be edited, digitized and then disseminated over the Internet or through CD-ROM.

Digital movies are typically expected to be large data structures with poorer quality reproduction than analog tape or video-disk. However, for computer demonstrations this is not the case, and movies may be made that are accurate in reproducing the details of the computer screen while occupying little disk space. This section details some practice in experience in producing such a movie with a resolution of 512 by 342 pixels, high-quality sound, playing for 8 minutes and occupying only 11MBytes. Figure 3.1 shows a frame from the movie being played in Apple's *MoviePlayer*. It shows interaction in a word processor with a document that itself contains an embedded QuickTime movie. The size and resolution of the demonstration are apparent in the quality of the text and images shown.

3.1 Topic of the Movie

The movie is a tutorial demonstration of knowledge acquisition, active document and expert system technology developed in the KSI, all technologies that are relevant to GNOSIS and were demonstrated at various workshops. It was originally prepared for the CD-ROM publication of the ACM SIGGRAPH MultiMedia93 Conference, Anaheim, California, August 1993, and is a demonstration of technologies described in a paper in the proceedings (Gaines and Shaw, 1993).

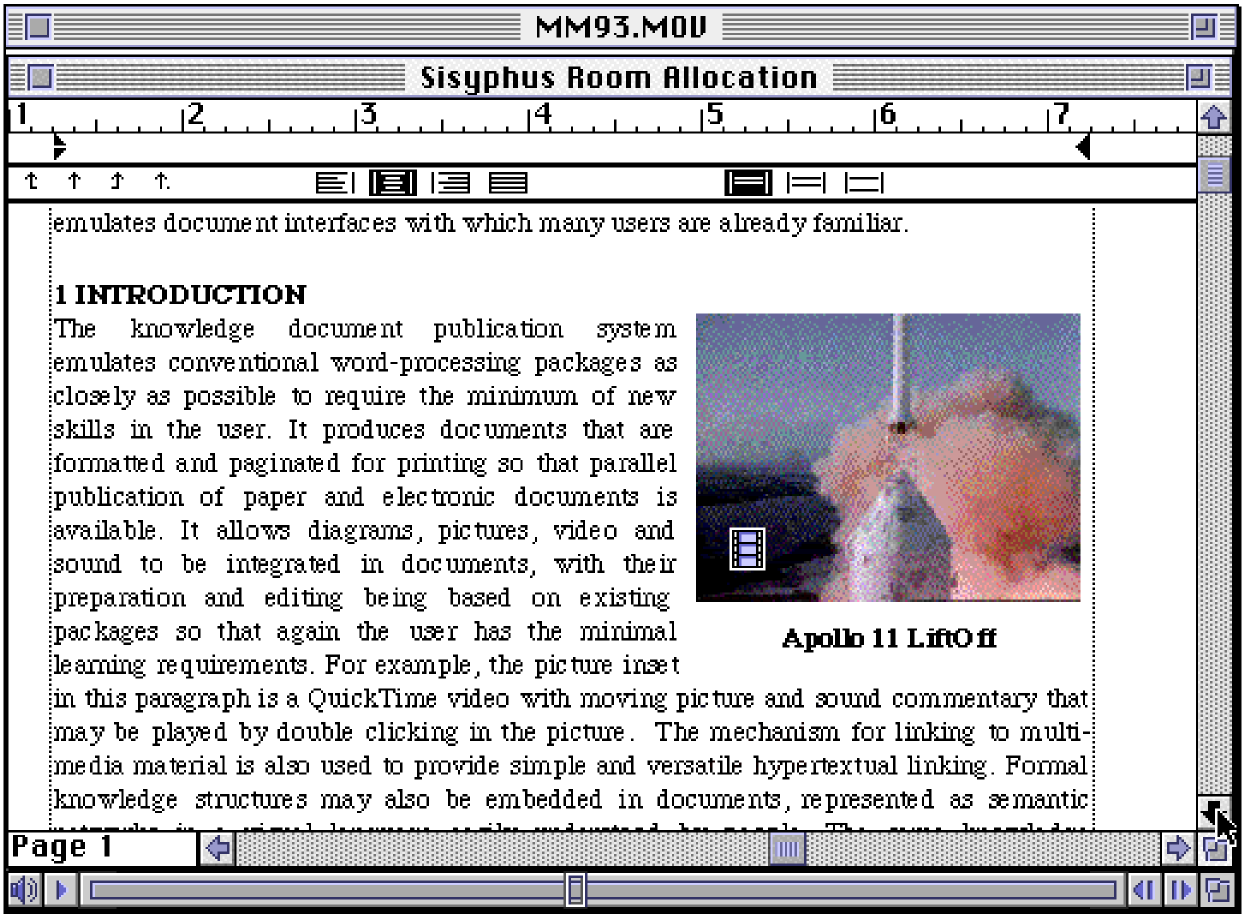


Figure 3.1 Tutorial demonstration as a high-resolution QuickTime movie

The movie demonstrates *KWrite*, an open-architecture, multi-media, active document production system, being used to develop, publish and use an active document in which the diagrams are knowledge structures in the form of semantic networks. They can be edited from within the document and interrogated by other applications as the knowledge base of an expert system. The document shown in the movie is an example of the actual application of *KWrite*. It was published as camera-ready copy by Cambridge University Press in December 1992 in the Proceedings of the British Computer Society Expert Systems Conference (Gaines and Shaw, 1992). It was also issued for FTP on Internet and on CD-ROM as a parallel publication of an active document that could be used for solving room allocation problems.

3.2 Production of the Movie

The movie was intended not only to supplement the paper as a live demonstration of the research described, but also to illustrate the potential of Apple's *QuickTime* for supporting such demonstrations. It pushed the current *QuickTime* implementation to its limits by adopting a large format, 512 by 342, that enabled it to show accurate screen dumps of *KWrite* in action. Given that it plays for 8 minutes with continuous sound and large-screen color video, the size of the movie at some 11 Mbytes is reasonably compact. This is possible through the use of the animation compressor whose data format is well-suited to screen dumps.

The primary tool used was *Camerman* from Vision Software, which provides live screen capture from Macintosh computers. Draft sound overlays were added using *CamerManEdit* which allows one to add sound to a captured movie while it is playing. This is a good technique for generating a spontaneous commentary, but not for getting smooth voicing and good sound quality. Hence, the scripts generated this way were transcribed into text and re-recorded separately in a quiet room using a Casio *DA-R100* DAT recorder.

The *R100* is a high-quality machine, small enough to slip into a pocket, offering two 16 bit 34Kbytes CD-ROM quality audio channels with 90db signal/noise ratio and no cross-talk between the stereo channels. It is useful for conventional movie capture to supplement a camcorder giving three channels of sound in the field, with one microphone on the camcorder and two on the R100. The peak-holding LCD level meters on the *R100* span 80 db and enable one to see the room noise accurately as well as setting microphone levels to avoid clipping.

The sound was digitized at a Macintosh *Quadra 900* through a *dbx 1* sound compressor. This is usually necessary with camcorder input from the field where sound levels may fluctuate quite widely. The *dbx 1* with its extensive level controls is also a useful front end to the computer since the line inputs are somewhat insensitive and result in unnecessarily low sound levels from standard line levels.

The compression was not all that important in this application because the scripts were recorded under controlled conditions. However, a dbx compressor is a valuable tool in attaining good sound quality in *QuickTime* digitization. With 8-bit sound it is worthwhile conditioning the signal on the audio side to be at maximum peak level before digitization. When 16-bit audio becomes common much of this conditioning can be done post-digitization.

The audio was digitized at 22KHz using Adobe *Premiere*'s audio input, and the data rate was reduced to 11KHz in the compression phase without noticeable loss of quality. *Premiere* was used to synchronize the audio and visual tracks, primarily by adjusting the speed of play of the movie segments.

There is an interesting stylistic problem in putting voice over animation as to whether the voice is the primary focus of attention and introduces the visuals—in which case vision should lag sound—or whether the movie is the primary focus, with the sound a commentary on the images—in which case sound should lag vision. There is no universal rule and it is important to adjust the time relation until it feels right, with vision leading sound in most cases since this leads to the most rapid attention switches.

The synchronization problems of the current *QuickTime* implementation can override these subtle considerations, and it is worthwhile to try and detach the commentary somewhat from the visual changes. On slower machines the sound can gradually come to lag several seconds behind the video. To compensate for this the movie was divided into 6 segments and a silent period of some 5 seconds left in the sound track between sections. This allows *QuickTime to* resynchronize the sound and movie between sections.

The six segments are: an introduction, an explanation of the *KWrite* document architecture, a demonstration of the knowledge elicitation tool *KSS0*, a demonstration of the knowledge editing tools *KDraw*, a demonstration of *KWrite* representing the knowledge embedded in a document, and a demonstration of a *HyperCard* stack being used to solve a room allocation problem by interrogating the knowledge base in the *KWrite* document.

The four demonstration sequences were captured by *CamerMan*. The two initial animation sequences were developed as sequences of PICTs in *Canvas*, loaded into *Premiere* and adjusted in length to synchronize with the commentary.

The six segments were pasted together in *MoviePlayer* and then compressed into a single fork movie with a maximum 90KBytes/sec data rate in George Cossey's *MovieShop*. The single-fork 'flattening' is important in allowing it to be played under Microsoft *Windows* on PC platforms. It is important in using *MovieShop* to compress animations to set the 'similarity level' at which it discards frames higher than for camera video so as to avoiding losing small changes that are significant to the animation. The default is 240, and 252 was appropriate for the animations in this movie. *MoviePlayer* and *MovieShop* were issued by Apple as part of the original *QuickTime* distribution. *Premiere* 4.0 now has both data rate limitation and flattening capabilities.

The movie was transferred to CD-ROM using a Philips *CDD521* recorder and OMI *QuickTOPiX* software. It was also put up for FTP on the Internet.

The movie demonstrates that tutorial demonstrations are readily created and disseminated as digital movies. Production is neither complex nor expensive—*CamerMan* alone at under $100 could be used to produce virtually all of this movie.

4 CAPTURING A WORKSHOP IN A MULTIMEDIA PROCEEDINGS

Formal workshops and conferences conventionally put much effort into producing a proceedings in a standard format specified to the authors so that they can produce camera-ready copy. However, modern optical character recognition (OCR) software coupled with style sheets in a word processor can be used to create a high-quality proceedings from original documents in arbitrary and varied formats. In addition, presentation slides can be digitized and presentations and discussion videotaped and digitized to create multimedia material that can be included in the electronic version of the proceedings.

As an example, a camcorder was used to capture software demonstrations and presentations at a 3-day GNOSIS workshop in Tokyo in March 1993 involving some 30 participants. The slides presented were collected together with the handouts circulated, largely technical papers from conference proceedings and internal reports. On return to Calgary, the papers were optically recognized using Xerox *AccuText* with an accuracy of some 98%, remarkably high considering that some of the papers were n'th generation copies of double-column conference material in 9pt type. The slides were digitized and the movies converted to QuickTime and lightly edited in Adobe *Premiere*. The movies were shot with a Sony *TR101* which has optical image stabilization that compensates for the slight instabilities of long hand-held shots. The editing was necessary primarily to allow for participants walking in front of the camera in what was a very informal setting—it is not appropriate to attempt to establish studio conditions at a scientific workshop.

The resultant material was put into a uniform style in Microsoft *Word*, resulting in a 300-page workshop proceedings (GNOSIS, 1993) containing 30 articles and 8 QuickTime movies. Figure 4.1 shows a page from the proceedings with the heading, comments, and movies of computer software demonstrations preceding a technical paper. The production time to OCR the source material, edit it into a uniform format, digitize the slides and movies, and issue the proceedings in print and CD-ROM was one week. The effort required was some 40 person-hours. Thus, a high-quality workshop proceedings can be produced *after* the event without any additional effort on the part of the participants, indeed without any advance warning that there is any intent to do so. The total cost of producing the proceedings was about $3,000, about the same as the air fare to travel to Tokyo.

This application is significant to the support of the semi-formal discourse in scholarly communities that takes place through seminars, workshops and informal conferences. Much of what occurs at such meetings in currently lost and not disseminated to others, and yet digital document technology now makes it simple to capture the essence of a meeting without imposing constraints on the participants.

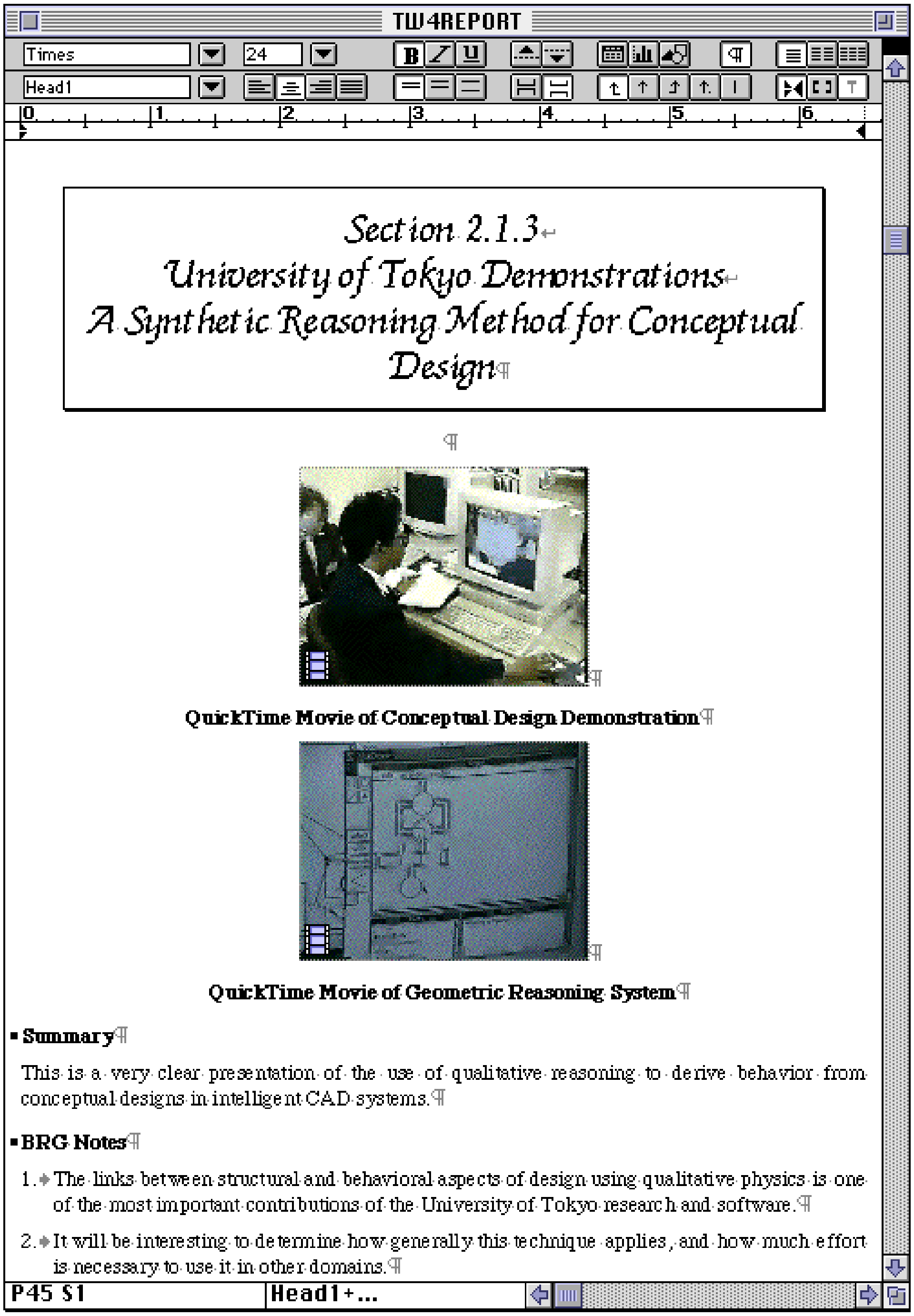


Figure 4.1 Workshop proceedings with embedded demonstrations

5 ARCHIVING A RESEARCH PROGRAM ON CD-ROM

At the end of the first year of the GNOSIS project, the project reports, slide sequences, photographs, movies, and so on, were issued on a CD-ROM. To enable widespread access to the reports, a hybrid format was used that could be read on Macintosh, Windows and Unix platforms, and all the reports were issued in Microsoft *Word*, Farallon *Replica* and Adobe *PostScript* formats. The total volume of material on the CD-ROM was: 57 reports totaling 1590 pages, each in 3 formats; 11 movies totaling 70 minutes; plus software and maps of the material. 3 test masters were made of the CD-ROM using a Philips *CDD521* recorder and OMI *QuickTOPiX* software. The third one was sent to a CD pressing company that produced 500 copies in 10 days at a cost of $1500. The second test master was very close to the final one and was sent off at the same time to a Japanese partner in GNOSIS who was scheduled to demonstrate the GNOSIS materials at a major IMS conference shortly after the production versions would be available.

The combination of high-speed production of one-off CD-ROMs and low-cost production of quantity CD-ROMs provides an extremely powerful and flexible technology for the dissemination of technical material. In particular, the space required to archive CD-ROMs is very much less than that of the equivalent paper documents, and the cost of mailing them is also very much lower. Additionally, the CD-ROM can contain colored diagrams, photographs, sound movies, simulations and other software, that it would be very much more expensive to issue in other formats.

5.1 Read-Only and Speed Constraints

The various technologies for digital document and CD-ROM production are now mature enough to be highly significant in the issue of multimedia materials. However, they are changing very rapidly and it is probably useful to go through the stages of production, the decisions made and technologies used at each stage, the details of what has already changed and the forecasts of what is expected in the near future.

The CD-ROM technology in itself is extremely simple. A CD-ROM for the Macintosh is a bit-by-bit image of a normal magnetic drive and hence appears to the user as a disk drive whose only peculiarities are that it is read-only and somewhat slower than a magnetic drive—typically 150KBytes/sec for a so-called ‘single-speed’ drive, 300KBytes/sec for ‘double-speed’ and so on. To a large extent one can treat the preparation of a CD-ROM exactly as if one were loading the document archive onto a standard magnetic drive.

The ‘read-only’ limitation means that one has to be very careful to prepare the CD-ROM accurately. Errors cannot be corrected, although it is possible to ‘update’ a CD-ROM by using software that checks an associated magnetic drive for updates before it reads files from the CD.

The speed limitation (compared with a typical magnetic drive speed of 1.5MBytes/sec) means that one has to be very careful to avoid multiple disk accesses in fetching one item of information. ‘Defragmentation’ software needs to be run before an image of a magnetic drive is transferred to CD-ROM so that directory and file structures are in contiguous segments and not scattered on different tracks. This can make the difference between an access of 1 second and one of 30 seconds which greatly affects the usability of the CD. There are also more subtle impacts on speed such as the need to demount a hard drive on a Macintosh before transferring its image to CD-ROM. This ensures that the status information about the drive indicates that the directory structure is correct and avoids a long delay while the Macintosh operating system checks this when the disk is mounted. Good quality CD-ROM preparation software checks for fragmentation and demounts the disk *before* it commences the irrevocable process of writing to a one-off CD-ROM.

Another impact of the speed limitation with multimedia material is the need to ensure that it can still be read effectively at the lower data rate of the CD-ROM. As noted above, programs such as *MovieShop* and *Premiere* have options to filter a digital movie to reduce the data rate required to play it to a sufficiently low value to enable it to be played from a CD-ROM. The typical data rate specified for a single-speed CD is 90KBytes which gives time for both data access and movie decompression. The filtering essentially reduces the effective frame rate and lowers the quality of reproduction, but this is far preferable to the erratic stop-start playing of both sound and video if the data rate is inappropriate to the media.

5.2 Multi-Platform Formatting

CD-ROMs for the Macintosh, PC and Unix platforms can each be based on bit-by-bit images of the file structures on the standard magnetic drives for the respective platforms. However, these structures are very different between platforms so that some other approach is necessary to create a single CD-ROM that can be used on all three platforms. One approach is to produce a CD-ROM in ISO 9600 format which closely resembles the MS-DOS file format and can be read by CD-ROM drive software on all three platforms. This is a common format for many cross-platforms CD-ROMs containing largely textual data such as programs.

The problem with the ISO 9600 format is that Macintosh users in particular are used to accessing their disks through a the spatial layout of an iconic ‘desk top’ with descriptive file names in upper and lower case, and these capabilities are lost in the MS-DOS 8 character name plus 3 character extension, uppercase file naming format. Fortunately, it is possible to produce hybrid CD-ROMs whose directory structure appears as a normal desk top database to the Macintosh system, and as an ISO 9600 MS-DOS name structure to the DOS, Windows and Unix operating systems. This possibility is not a design feature of the operating systems but rather a convenient ‘hack’ made possible by the differing conventions of where the different disk drive software expects to find directory information.

5.3 Portable Document Formats

The choice of document formats for the issue of archives on CD-ROM is not an easy one. If the documents are to be reused it is helpful to issue them in a common word processor format such as Microsoft *Word* or Novell *WordPerfect*. Most readers will have a word processor that either reads such documents directly or can import them while preserving most formatting information. However, word processed documents are designed for editing and reformatting, and do not preserve their pagination across different platforms and paper sizes. One rarely obtains an identical document when one exports between word processors, across the same word processor on different platforms, or between countries using different ‘letter’ paper sizes. There is also the problem that the word processors commonly used on the Macintosh and PC are rarely available on Unix workstations.

These problems have prompted the development of new technologies and products for portable electronic documents that can be read and printed across all platforms. The concept is to provide a system that:

- allows a document to be ‘printed’ to an electronic file in a portable document format that preserves appearance and pagination;
- provides a simple viewer for each platform that allows such documents to be viewed, printed and, possibly, searched by content and clipped for pasting to other documents;
- provide more complex document management systems that allow such documents to be indexed, hypertext linked, repaginated, and so on.

There are now a number of products that support this concept, notably Adobe *Acrobat*, Farallon *Replica*, No Hands *Common Ground* and Novell *Envoy*. They vary greatly in maturity, capabilities, royalty arrangements, and the state of both products and the nature of market is in flux (Seybold, 1994). Portable document support is seen as a growth market but it has been very difficult for commercial organizations to find a product and royalty structure that matches the needs of customers and is profitable. For example, when Adobe introduced *Acrobat* it was first to market but required payment for even the simplest viewer as did No Hands for *Common Ground*. This limited the utility of sending out documents on the Internet or CD-ROM since there was no guarantee that the user would have a viewer and it was expensive for the document supplier to provide one. Farallon overcame this with *Replica* by offering a free viewer and putting it up for FTP on the Internet for ease of access. No Hands rapidly followed suit, and Adobe has done the same with its release of *Acrobat 2*. Novell has done something similar with *Envoy* but only supports a free viewer integrated with a document which limits cross-platform portability—one has to prepare and issue one document for the Macintosh and one for the PC.

For CD-ROMs for which payment is made, it is attractive to consider the extended viewers which offer various forms of indexing, hypertext linking, document corpus management, and extension capabilities through application program interfaces (APIs). The cost of the associated tools for document preparation and the royalty arrangements for issuing the viewers on CD-ROM vary greatly, and are changing with time. It is necessary to check the current state of each vendor’s product each time one makes a CD.

On more complex material, such as slides from a presentation package such as Adobe *Persuasion*, it is important to check the quality of document conversion both on the original platform and cross-platform. The usual mechanism for conversion to a portable document is to insert a specialist ‘printer driver’ that emulates the actual printer driver on a particular platform and operating system. The document generation process then consists simply of ‘printing’ to the specialist driver. However, it is not a simple task to capture complex color layouts into a portable document format from an arbitrary range of packages which have in common only that they can produce printed output. The existing products still have ‘glitches’ in the conversion of some material, and careful quality checking is required.

There are also other alternatives for portable documents. Frame Corporation’s *FrameMaker* has an output format that is portable across Macintosh, PC and Unix platforms, but its royalty costs for a reader are currently high and one is restricted to documents in *FrameMaker*. The Standard Generalized Markup Language (SGML) makes provision for Document Type Definitions (DTDs) that allow structures in documents to be marked up in a way that supports portability not only across platforms but also across different formats and media (Bryan, 1988; Goldfarb, 1990).

The HyperText Markup Language (HTML) used for World Wide Web documents is a DTD in SGML (Berners-Lee, Connolly and Muldrow, 1994; Raggett, 1994), and it is attractive to use it as a portable document format on CD-ROMs also since there are public domain browsers available for all platforms.

Adobe *PostScript* is also a useful cross-platform document format since it can be produced by most document systems, printed by most laser printers, and there are public domain browsers for all platforms. There can be portability problems with arbitrary *PostScript* files produced on different machines using different software, but portable *PostScript* is readily produced in the same way as the other portable document formats by the use of proper tools and careful quality checking. Its primary disadvantages are the comparatively large size of the files produced and the lack of search and clipping capabilities. The specially formatted and compressed *PostScript* of *Acrobat* is intended to be the preferred alternative but, until it is in widespread use, raw *PostScript* is still useful.

When the CD-ROM described in this section was produced, Farallon *Replica* had the only royalty-free viewer, and documents were issued in *Replica*, *Word* and *PostScript* formats. The seven final reports occupied 2.3 MBytes each in Replica and Word, and 4.7 MBytes in PostScript. Thus, the storage required for all three formats is about four times that for Word or Replica alone. The storage requirement was not significant in this application because the 1590 pages of reports take up only some 10% of the CD-ROM capacity when issued in all three formats. Video material usually establishes the dominant requirement for storage.

5.4 Indexing

One factor significant to the usability of a CD-ROM is the type of indexing provided. It is also very relevant to the CD-ROM as a presentation medium. The ‘corporate image’ projected by access to material through a simple Unix or Windows directory is not the same as that projected by access through some multimedia navigational tool that provides a user-friendly contextual map of the material. There are commercial tools that index document collections by content and provide an information retrieval interface to the material. These are appropriate to reference materials that need to be searched by content. However, they again present a rather ‘technical’ and unattractive interface, and even when they are used it is appropriate to provide higher-level maps of the material.

Apple’s *HyperCard* on the Macintosh and Asymetrix’s *ToolBook* on the PC provide a basis for developing attractive multimedia access to a CD-ROM of multimedia materials. Both products support graphics, color, sound, video and direct manipulation interfaces. They are able to open files in other applications such as *Word* or *Replica*, and hence can be used to access heterogeneous materials using the native application. While the products do not operate cross-platform, is possible to develop stacks in *HyperCard* that are virtually identical in appearance and functionality to those in *ToolBook*, and vice versa, so that a users can be largely unaware of cross-platform variations.

For the GNOSIS archives, it was appropriate to use as an indexing tool *Mediator* (Gaines and Norrie, 1994), a system that had been developed to support collaborative activities across the network as part of the GNOSIS research program. The *Mediator* implementation was based on groupware concept-mapping tools that were already in use for indexing multi-media materials (Gaines and Shaw, 1994a). Figure 5.1 shows the GNOSIS project archives being accessed through layered concepts maps. The map in the window at the upper left is a top level "Server Agent" that manages a particular collection of material. In the example shown a local user is accessing material directly through this agent. Remote users connect to the server agent over the network using client agents that give them the same functionality through calls to the server.

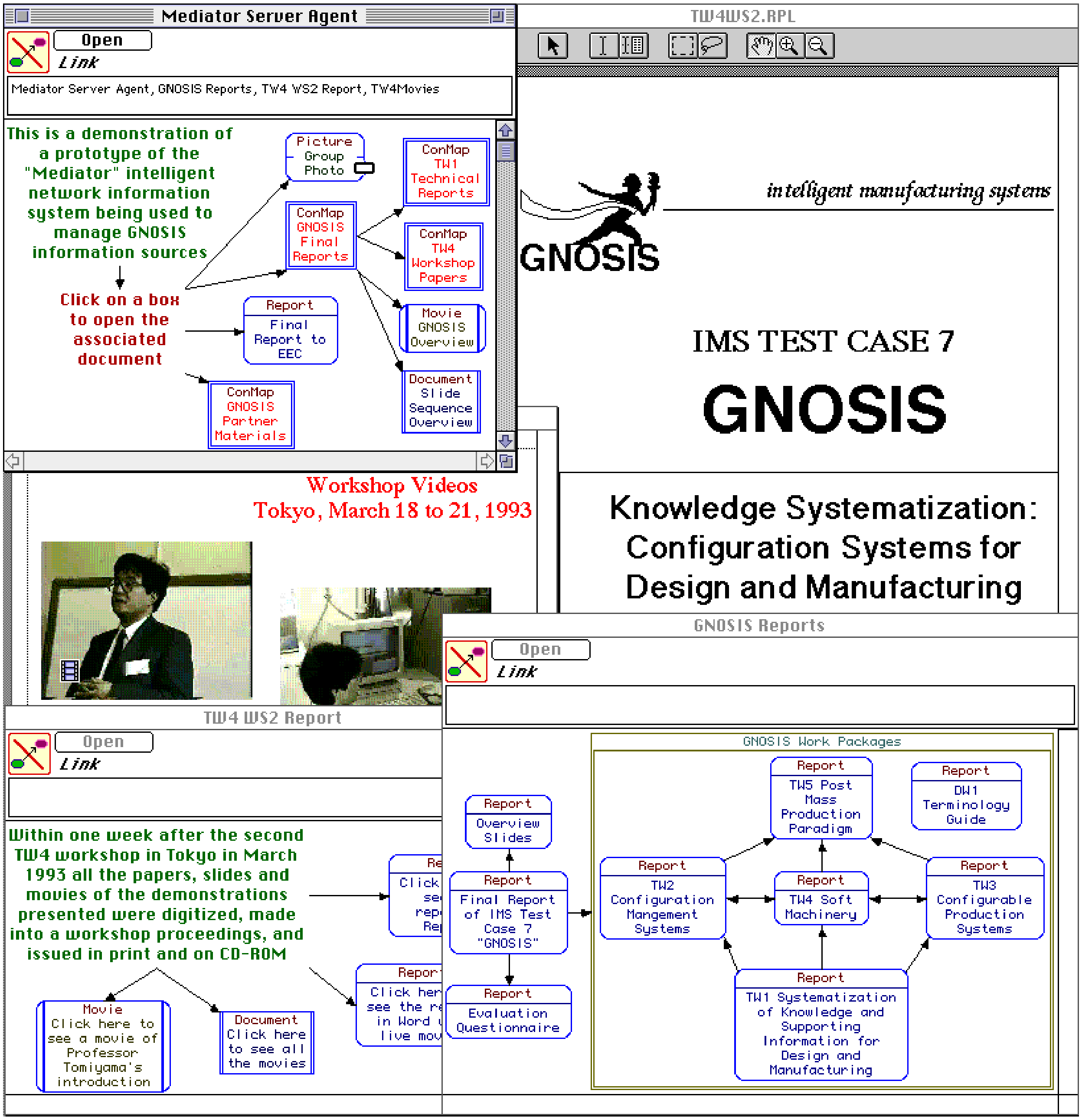


Figure 5.1 Accessing the GNOSIS archives through layered concept maps

The concept map at the top left is currently write-disabled, and the cursor has changed to a button as the user mouses over the "Group Photo" node. Clicking at this point will display the photograph in a separate window. The user has already clicked on the node "GNOSIS Final Reports" to open the concept map shown at the lower right. This has a node for each report, and clicking on one will open the appropriate report, in this application using Farallon's *Replica*. The node at the top left gives access to a series of slides on the project displayed using *Replica*. A similar node in the original concept map at the top left gives access to a movie on the GNOSIS project that will be opened in Apple's *MoviePlayer*.

The user has already clicked on the "TW4 Workshop Papers" node in the concept map at the top left, and opened the relevant concept map at the bottom left. She has then clicked on the node "Click here to see the report in Replica", and opened the report visible at the back on the right of Figure 5.1. She has also clicked on the "Click here to see all the movies" node and opened the KWrite document visible behind the concept maps. This displays eight QuickTime movies of various demonstrations given at the Workshop, any of which can be played by double clicking on it.

6 COLLABORATION THROUGH THE INTERNET

The interconnection of large numbers of computer networks across the world to form the Internet has provided a communication medium supporting many forms of discourse in geographically dispersed communities. Since digital networks are content-neutral, they provide the means for disseminating multi-media of arbitrary type in a wide variety of formats. Since the networks provide access to digital storage resources, they provide the means for establishing on-line archives of multimedia material. Since the networks operate in real-time, they provide the means for a wide variety of forms of interaction. Since the networks also provide access to computing resources, they provide the means for offering a wide variety of computational services from those for indexing and information retrieval, through those for data analysis, to those for intelligent agents.

The growth of public interest in the Internet has given rise to a proliferation of guides to the net and its services. Krol's (1992) *Whole Internet* is a good starting point for those setting up services for a professional community. The following sub-sections give an overview of Internet facilities relevant to supporting collaboration.

6.1 Discourse on the Net—Electronic Mail

The foundation for many other services on the Internet is the capability to send *electronic mail* (email) from one individual to another. This is possible because each machine on the net has a unique address, and each person with an account on that machine has a unique name on that machine. Thus, point to point transmission can be supported in which a digital message is sent to a specific individual and placed in their digital mailbox. Because the message is digital it can comprise not only text but also multimedia attachments such as pictures and sounds. Because the addressee is a computer file it is possible for the effective addressee to be a computer program that takes action based on the mail, thus providing a basis for many services.

Mail readers provide a user interface to the mail system and a wide range of facilities for sorting, filing, indexing and filtering mail. The serious use of email requires the use of a mail reader with such facilities if a user is not to become overloaded in managing their mail. Figure 6.1 shows POPmail on the Macintosh being used to access some person-to-person mail relating to GNOSIS communications. At the top are a set of function icons for replying to mail, saving it in a separate file, fetching it from the server to the local machine, printing it, initiating a new item, recycling it to one of a set of user-named archives, and returning to the current message browser. Beneath these is a scrolling list of messages indexed by source, subject and date, and sortable on any one of those keys. The selected message is shown in the scrolling window below. It consists of a from, date, to and subject header, followed by the message. The indented portion of the message was automatically inserted by the “Reply” function so that some portion of the message sent can be used to provide a context for the reply. The three ‘signature’ lines at the bottom are also routinely inserted.

POPmail provides facilities for searching mail by content, for archiving it under user control, and automatically unpacking other files sent as attachments to the mail. More complex mailers such as *Eudora* also provide facilities for automatically filtering mail to different archives by source, subject line or content.

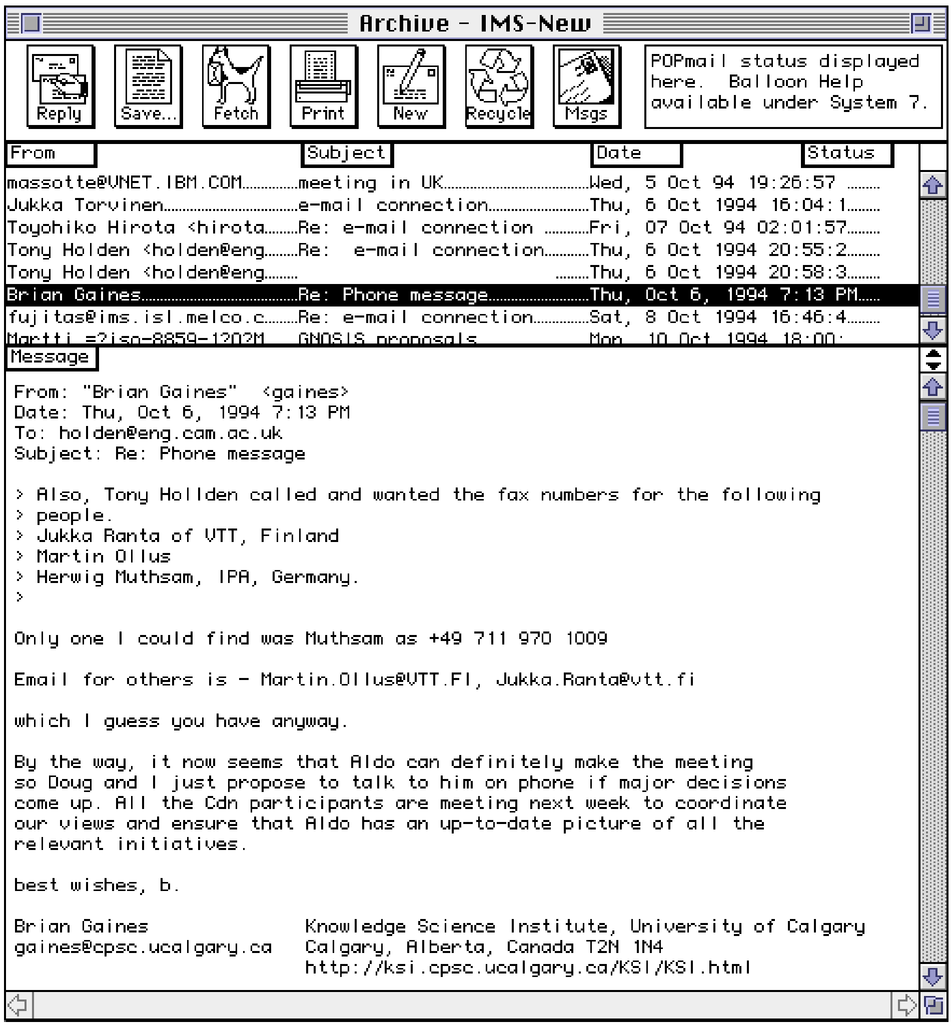


Figure 6.1 Accessing mail related to GNOSIS archived through a mail reader

6.2 Discourse on the Net—News

The *News* facility on the Internet provides a parallel service to email in which messages are not sent to an individual but to a community, again with a unique name but this time specifying a news group. Messages sent to a news group are distributed to all sites on the net that maintain news archives. Users at each site access these archives to read news items in much the same way as they access their mail boxes to read email. Multimedia attachments can be incorporated in news messages but this is not normally done since the volume of news is already a burden on networking resources—it is better to provide an address from which such material may be fetched using the file transfer protocol. Because of the volume of news and the way that is moved from site to site, it may take several days for a news message to reach a site remote from the sender, which limits its utility in supporting spontaneous discourse.

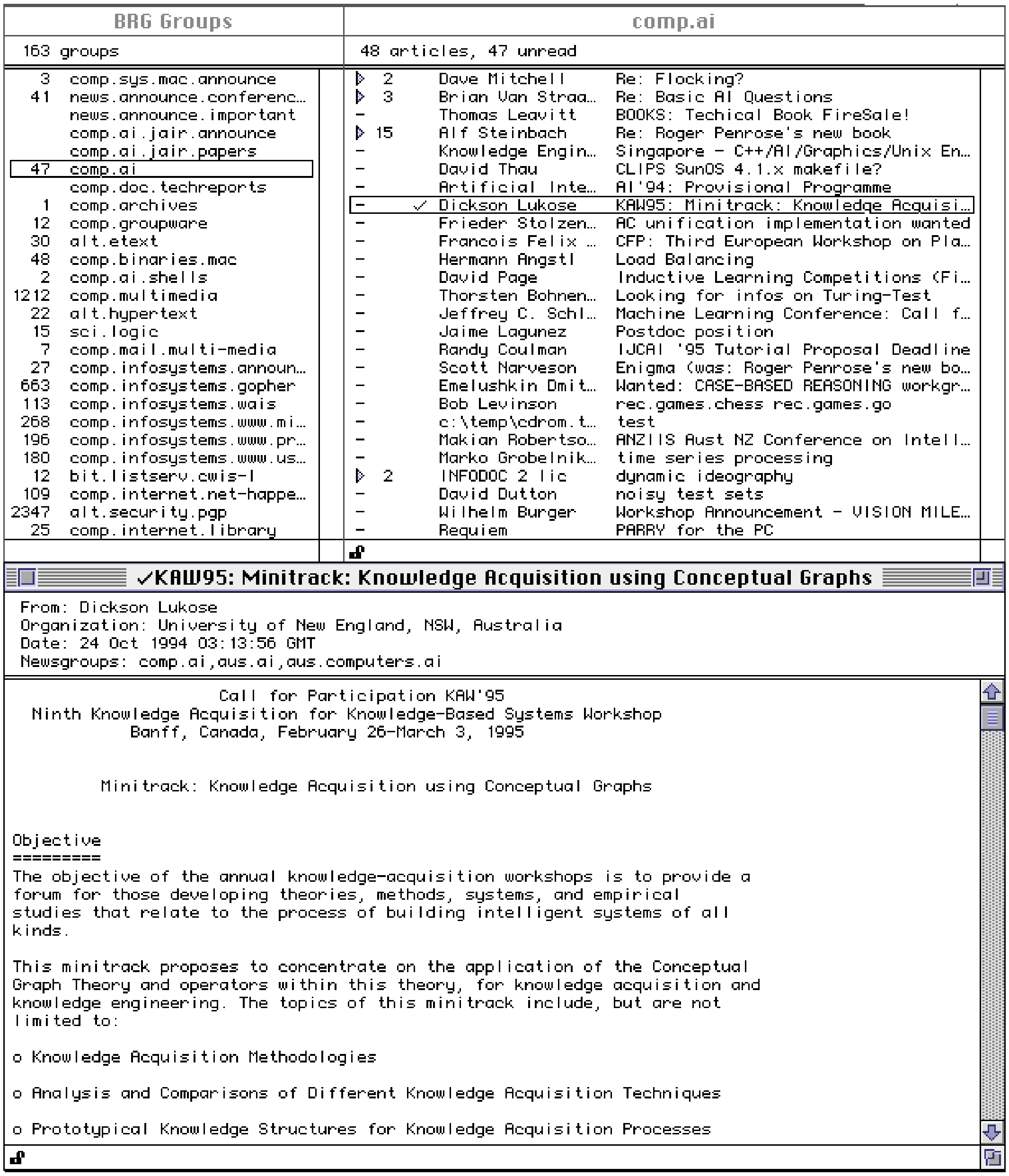


Figure 6.2 Accessing News groups through a news reader

The volume of news is very high with many groups receiving several hundred messages a day, and it is again important to use a reader that provides an excellent user interface for managing access to news. Figure 6.2 shows *NewsWatcher* on the Macintosh being used to access an item in the group comp.ai. The window at the top left lists the number of items unread in a list of user-specified news groups that interest the user. The group comp.ai has 47 items and when it is double clicked the window to the right appears giving a list of those items by source and topic, grouped by the threads of initial item and related replies. When the item posted by Dickson Lukose is double clicked it opens in the window at the bottom of the Figure, giving information on a conference relevant to GNOSIS's knowledge systematization theme.

It is important to use a news reader that gives easy access to the news tracks through simple browsing mechanisms that support selective browsing by relevant groups and by content. Internet News is a valuable resource but the sheer volume of news can be overwhelming if the user interface is inadequate. Good news readers provide simple mechanisms for printing and filing interesting news items similar to those of a mail reader.

6.3 Discourse on the Net—List Servers

Email supports private individual discourse, and News supports public broadcasting to major communities at large. *List servers* provide an intermediate service supporting selective dissemination of private information to specific communities. The principle is simple—an email message is sent to an address which is not that of a person but rather that of a list server which resends the mail to a list of email addresses maintained on the list server machine. To the person sending the message it presents a convenient way of sending mail to all members of a specific community. However, many benefits accrue from the use of a list server:

- the membership list is maintained in one location so that individuals do not have to keep track of who is in the community;
- becoming a member is usually supported through automatic processing of a 'subscribe' message sent to the list server;
- membership can be controlled, if required, by a human moderator so that messages are kept private to a well-defined community;
- all messages sent to the group can be archived and indexed so that the list server archives come to act as a 'corporate memory' based on discourse within the community.

Since list servers use electronic mail, access to them is through the same mail readers as used for person-to-person email. Figure 6.3 shows POPmail being used to access mail sent by one partner to all partners through the list server. The only difference from the person-person-mail of Figure 6.1 is that the "To" field in the mail itself specifies that it is to "gnosis-all@IPA.FhG.de" which is the address of a list server at the Fraunhofer Institute in Germany. The list server resends the mail to all those GNOSIS partners on its list.

Setting up a list server has proved to be a very effective way of supporting collaboration in geographically dispersed communities through the Internet. However, while the principle of resending mail to a list is very simple, there are many pitfalls for the unwary, and it is important to use a well-designed list server of which many are now available. A good list server incorporates techniques to:

- filter out subscribe and unsubscribe messages sent in error to the entire mailing list rather than the administrative component of the server;
- avoid loops in which an error or 'on holiday' message is automatically sent back to the server, resent to the originator generating another reply, and so on;
- allow easy subscription and unsubscription, and its control if required;
- support easy human moderation of messages sent, if required;
- archive messages and support controlled access to archives, if required.

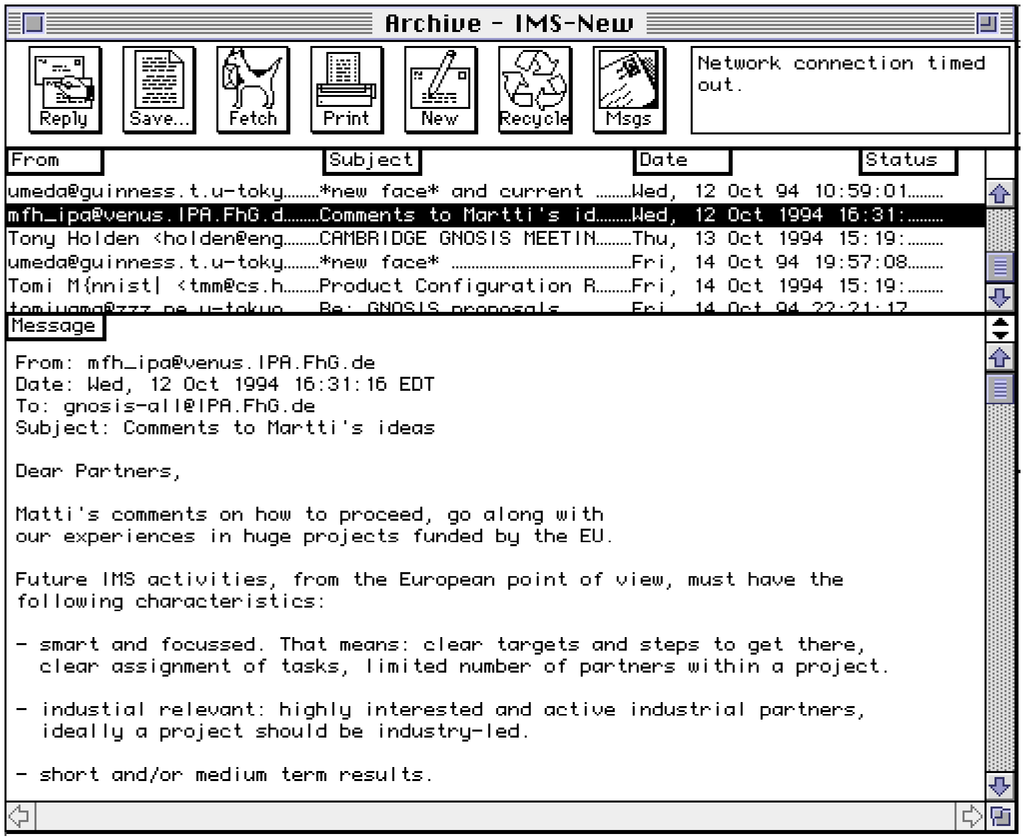


Figure 6.3 Accessing mail through a GNOSIS list server

6.4 Real-Time Discourse on the Net—Internet Relay Chat

The speed of Internet communications between many sites is such that electronic mail reaches its destination within a minute or so of being sent, and hence some degree of real-time, ‘conversational’ interaction is supported. In some situations, an even more rapid ‘conference’ mode of interaction is desirable, and this is supported through a number of *chat* facilities.

Internet relay chat (IRC) is the major system supporting conferencing world-wide (Pioch, 1993). A conference participant runs an IRC client program on his or her workstation which communicates with one of the major IRC relay sites that coordinate communication across IRC world-wide. Anyone can set up an IRC conference by defining a named IRC ‘channel’, and specifying the terms of membership to be public or private to an invited group. One joins the conference through a simple command, sends messages to it a line at a time, and sees these lines from all participants appear in a log screen shortly after they are sent.

Figure 6.4 shows IRCLE, a Macintosh IRC client, connected to the #Macintosh conference on IRC which is used by Macintosh users on IRC to discuss problems, new software, and so on. Each user specifies a nickname which appears in angle brackets at the beginning of their messages to identify the source. As shown in the top window, the order of messages is chronological and multiple simultaneous conversations become intertwined which can be confusing but supports spontaneous interactions in a way that a more formal protocol would not. The small window at the bottom is where the local user types messages and commands.

IRC discourse can be logged by any user, and there are famous logs available on the net relating to Gorbachev's disappearance, operation Desert Storm, and so on. There are also interesting sociological studies of the cultures on IRC, behaviors of participants, and so on (Reid, 1991).

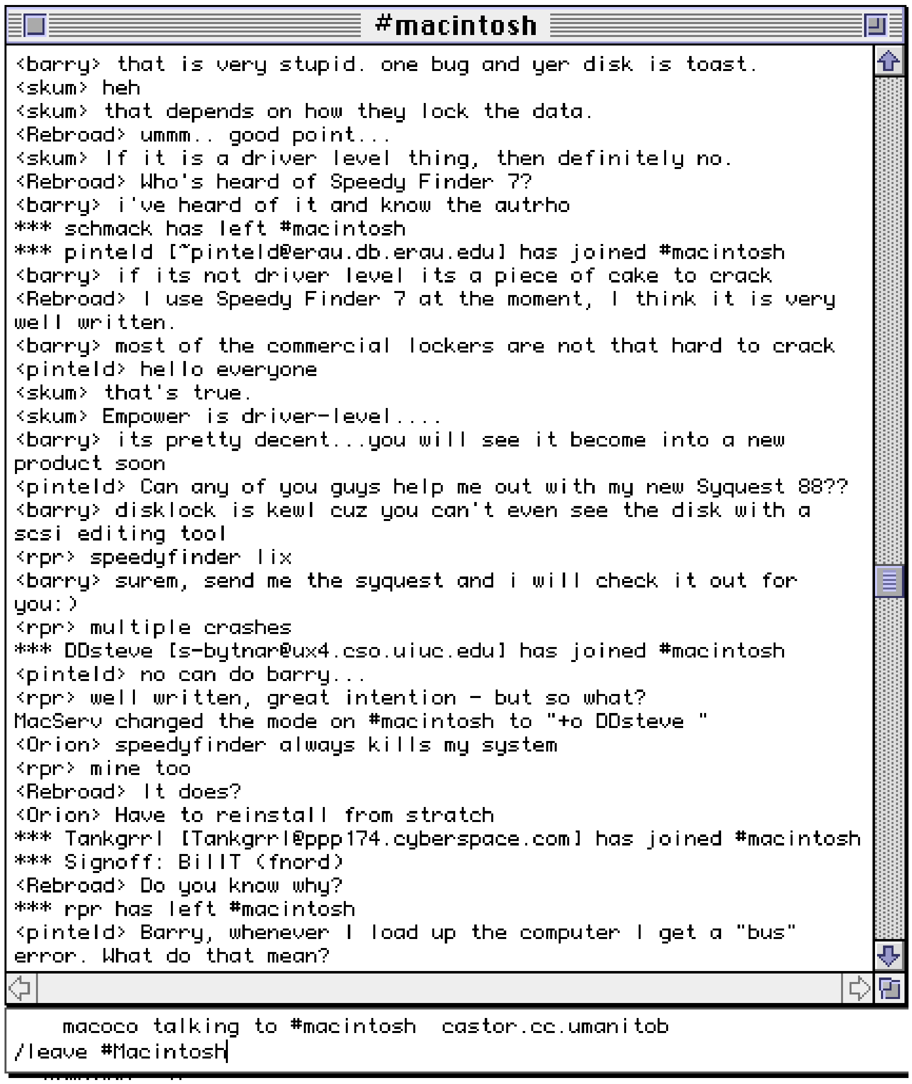


Figure 6.4 Real time discourse through Internet Relay Chat

6.5 Real-Time Discourse on the Net—MUDs

Multi-user dungeons (MUDs) are, as the name suggests, environments for playing fantasy games originating from dungeons and dragons. However, the technology is increasingly being used to support a variety of professional community activities (Curtis, 1993). The basic MUD concept is to provide a spatial metaphor for the discourse—the MUD equivalent of an IRC channel is a 'room'. The spatial metaphor extends naturally from participants to objects, and this provides a natural interface to other Internet services. A 'library' might contain a set of 'books' that can be 'read' by a participant. Asking what is in the room corresponds to requesting a directory listing. Asking to read a book corresponds to open the corresponding file in a text reader. The MUD provides a conceptual interface to Internet services which is natural in terms of every day experience. There are again interesting sociological studies of cultural formations in MUDs (Reid, 1994).

Astro-VR is a MUD for use by the international astronomical community:

> "The system is intended to provide a place for working astronomers to talk with one another, give short presentations, and otherwise collaborate on astronomical research. In most cases, this system will provide the only available means for active collaboration at a level beyond electronic mail and telephones. Initially, Astro-VR will provide the following facilities of interest to our user community:
>
> • real-time multi-user communication,
>
> • a self-contained electronic mail and bulletin board system,
>
> • shared, user-supplied *links* to online astronomical images,
>
> • an editor/viewer for short presentations of text and images,
>
> • collaborative access to standard programs used by astronomers, and
>
> • window-based shared editors." (Curtis and Nichols, 1993)

A MUD may be used as a convenient conferencing system by a collaborative group, such as a system support team:

> "We have found that the MUD is an effective way to hold pre-arranged meetings for people who can't be in the same physical location. We save a transcript of the meeting and email it to people who weren't present. It's common for us to have a five-minute conversation on the MUD about a small systems issue. Previously, these conversations would have happened through slower email, through office visits, or at regular systems meetings. All of these mechanisms are more cumbersome, and would have happened much less frequently. Thus the MUD has enabled new communications patterns." (Evard, 1993)

Because of their association with game playing, both IRC and MUDs are often neglected as resources for the support of professional communities. However, MUDs in particular offer an attractive way of managing access to Internet resources through a natural spatial metaphor, and they, or their derivatives, can be expected to play an increasingly significant role in the future. The main limitation currently is the text-based interface, and this is being overcome by the development of MUDs on the World Wide Web using the interactive color graphic interfaces available to the web.

6.6 Multimedia Archives on the Net—File Transfer Protocol

The availability of the Internet File Transfer Protocol (FTP) has been the major basis for supporting multimedia archives on the net. The protocol allows a user at one machine to request that files be transferred to or from any other machine on the Internet. To provide normal security a user must supply an account and a password to the remote machine corresponding to the capability to log in to that machine. However, it has become conventional to establish an 'anonymous login' account whose account name is 'anonymous' and whose password is symbolic which has access to a well defined sub-file system that is publicly accessible. The symbolic password requested is generally the email address of the user which can be used to monitor access. Some systems check that a syntactically correct email address is given, others that the domain corresponds to that of the calling machine, while others perform no checks at all. It is generally accepted that anonymous FTP access is open to all. The only controls are to restrict the number of simultaneous anonymous accesses allowed at popular sites whose Internet communications might otherwise become overloaded.

The Unix file transfer protocol is simple and it is supported by many different tools. For example, an email or news tool may support anonymous FTP access to files specified within email or news items. A good FTP tool will provide a user interface to email similar to file directory access on the local machine, a user-definable directory of commonly addressed sites, and facilities for unpacking files sent in one of the common compressed formats. Figure 6.5 shows Fetch on the Macintosh being used to access a sub-directory of the GNOSIS archives containing transcripts of presentations given by officials of the US Department of Commerce at an IMS meeting in Dallas. The directory of files on the remote machine on the left looks like a local Macintosh directory, file transfer is effected through simple "Put" and "Get" buttons, and the status of a transfer in progress is shown on the right.

In many scholarly communities large digital archives have already been established which fulfill the role previously filled by the circulation of paper preprints (Anderson, 1991). The high-energy physics archive contains over 10,000 papers and has become the major source of recent results for the associated community research community. Archives exist across all disciplines including poetry, philosophy, sociology, economics and mathematics. As well as providing easy circulation of preprints, they have also been the basis of successful 'electronic journals' that are not published in paper form (Gaines, 1993; Manitoba, 1993).

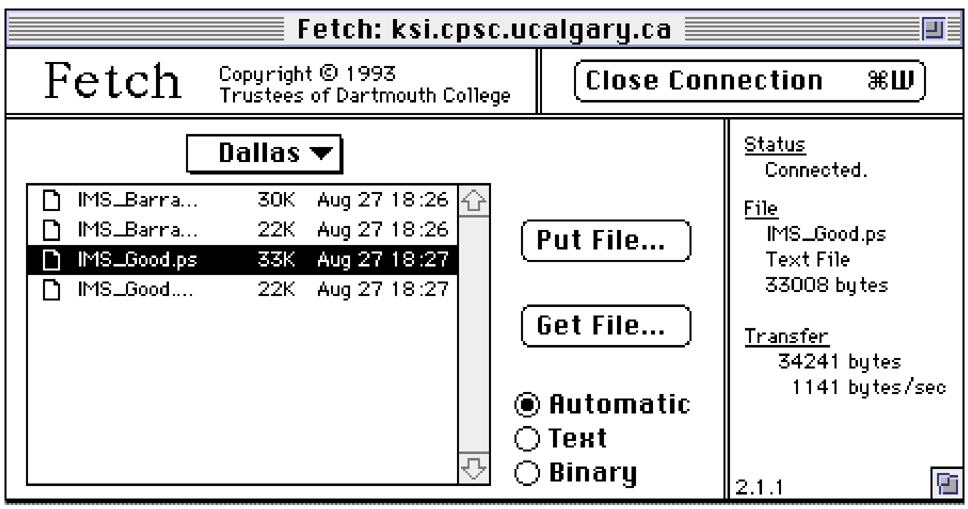


Figure 6.5 Access to document archives through the File Transfer Protocol

6.7 Indexing Archives on the Net—Archie

The growth of the digital publication and archiving on the Internet has been such that finding specific material is a major problem. There is no directory for all machines on the Internet, but it is common for individual disciplines to attempt to keep track of all the anonymous FTP archives relevant to that discipline. Such list are generally published through the relevant news groups or list servers. There are also tools that attempt to index material on the net in a variety of ways.

Archie is a system for indexing material at anonymous FTP sites and allowing users to search for specific documents by the content of their titles. There are a number of Archie servers world-wide that index different collections of machines. They may be accessed through a variety of tools which usually provide not only search facilities but also the capability to transfer and decompress files once their location is found. Figure 6.6 shows *Anarchie* on the Macintosh being used to search for files on STEP, and to access files on Ontolingua. *Anarchie* provides similar

FTP facilities to *Fetch*, and the window at the top is a user-defined list of commonly accessed FTP sites which may be accessed directly without search. The two windows below it are ongoing searches on two different servers for files whose name contains the string “STEP”. The bottom window is the result of a previous search for files whose name contains the string “ontolingua”. A directory and two files have been found at Stanford. An FTP transfer may be initiated by double-clicking on one of the file names.

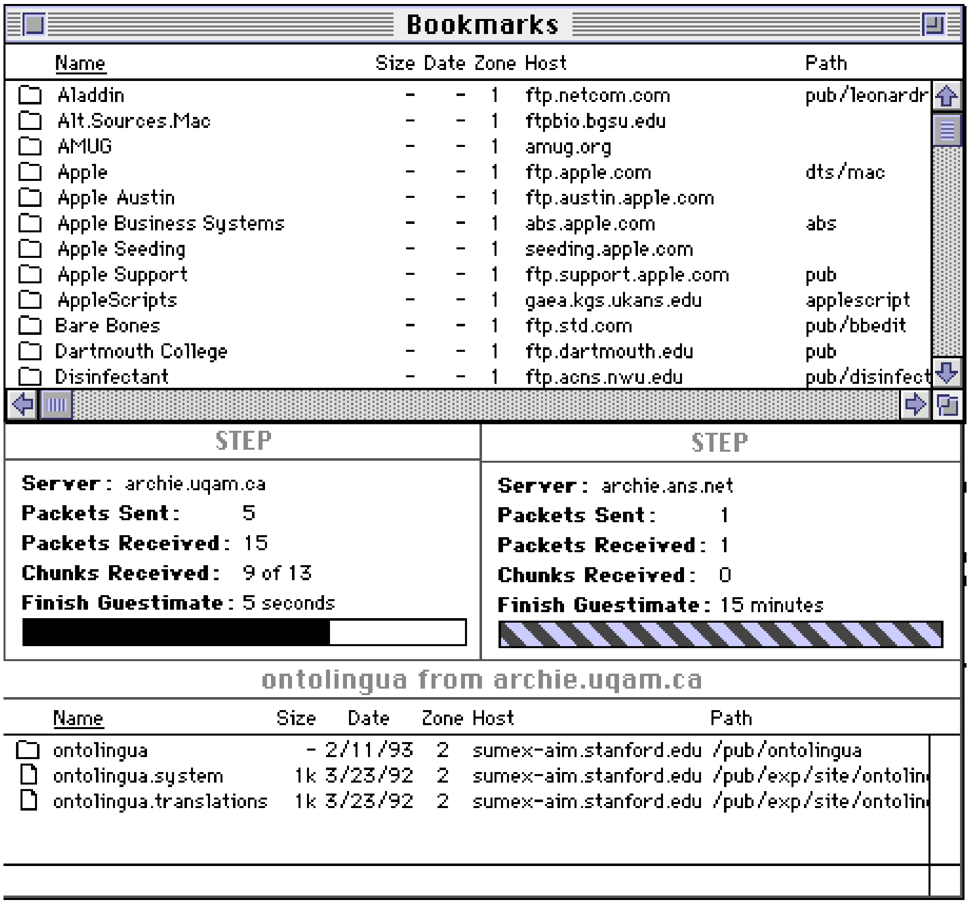


Figure 6.6 Searching document archives using Archie

6.8 Indexing Archives on the Net—Gopher

Gopher is a system for indexing material on the Internet not by name but by cataloguing it in arbitrary hierarchical indices. It was originally developed at the University of Minnesota to provide a campus-wide information system (CWIS), and such use is common across universities in North America. However, the Gopher cataloguing scheme proved so simple and yet so powerful that it has been widely adopted for indexing material on the Internet world-wide. The power of Gopher comes from the structure of its catalogs which are documents containing a title of a document, the type of document, the Internet address of the machine on which it is held, and a file path to the document on that machine. Document types include text, various graphic formats, PostScript, and Gopher catalog documents—this last providing the basis of hierarchical indices with parts scattered over many machines world-wide. Thus a researcher in France might maintain a Gopher catalog of manufacturing research reports from French and German research agencies. A researcher in Japan might set up an index to manufacturing research reports world-wide and include those in France and Germany by simply including a pointer to the French researcher's Gopher catalog.

Figure 6.7 shows TurboGopher on the Macintosh being used to access Internet archives. The window shown is the top level one at the University of Minnesota, and the origins as a CWIS are apparent. However by double clicking on the catalog item "Other Gopher and Information Services" one opens a second-level catalog of areas of the world that have Gopher sites. Double-clicking on an area opens a third-level catalog of sites, and so on until one reaches files which are fetched. Gopher has an interface to Archie which can be used to search as described above. The Gopher network is itself indexed by Veronica which can be accessed through Gopher to search for named items in Gopher space in the same way as Archie does for FTP space.

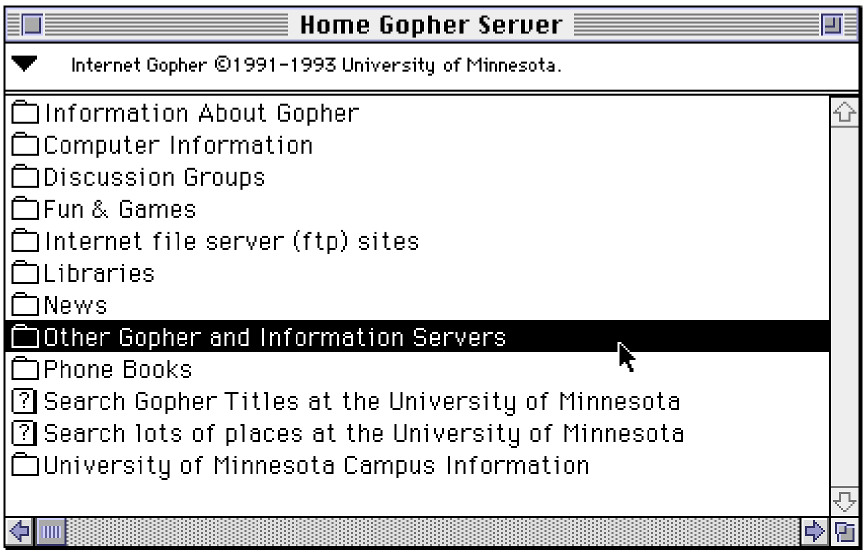


Figure 6.7 Accessing document archives through Gopher

6.9 HyperMedia Archives on the Net—World Wide Web

It will have been noted that the tools described have been cumulative in their functionality—Archie supports FTP, Gopher supports FTP, Archie and Veronica. The various Internet protocols and facilities are not competitive but complementary and cumulative. The ultimate cumulative system currently is the World Wide Web which subsumes all the protocols so far described, and adds to them hypertext links between active documents that support general client-server computing through graphic user interfaces.

World Wide Web was conceived by Berners-Lee in March 1989 (CERN, 1994) as a "hypertext project" to organize documents at CERN in an information retrieval system (Berners-Lee and Cailliau, 1990). The design involved: a simple hypertext markup language that authors could enter through a word processor; distributed servers running on machines anywhere on the network; and access through any terminal, even line mode browsers. World Wide Web today still conforms to this basic model.

A poster and demonstration at HT91 in December 1991 announced World Wide Web to the computing community. However, major usage only began to grow with the February 1993 release of Andreessen's Mosaic for X (Andreessen, 1993). Whereas the original proposal specifically states it will not aim to "do research into fancy multimedia facilities such as sound and video" (Berners-Lee and Cailliau, 1990), the HTTP protocol for document transmission was designed to be content neutral and is as well-suited to multimedia material as to text. The availability of the rich X-Windows environment on workstations supporting color graphics and sound led naturally to multimedia support, although the initial objective of meaningful access through any terminal was retained. Most web material can still be browsed effectively through a line mode browser.

Figure 6.8 shows MacWeb on the Macintosh providing access to the GNOSIS archives encoded as a hypermedia document collection. The GNOSIS final report has been fetched from a remote server across the Internet. It appears on the screen, and can be printed, with the typography, layout, and embedded colored diagrams expected of a high-quality document processor. It supports embedded hypertext links which can be used to access other documents across the net. For example, the GNOSIS logo near the top of the page in Figure 6.8 is an embedded picture. The underlined term "Section 8" at the end of line 3 of the Overview is a hypertext link. Clicking on the underlined term causes the document referenced to be fetched, in this case Section 8 of the report which itself has hypertext links to the other GNOSIS technical reports.

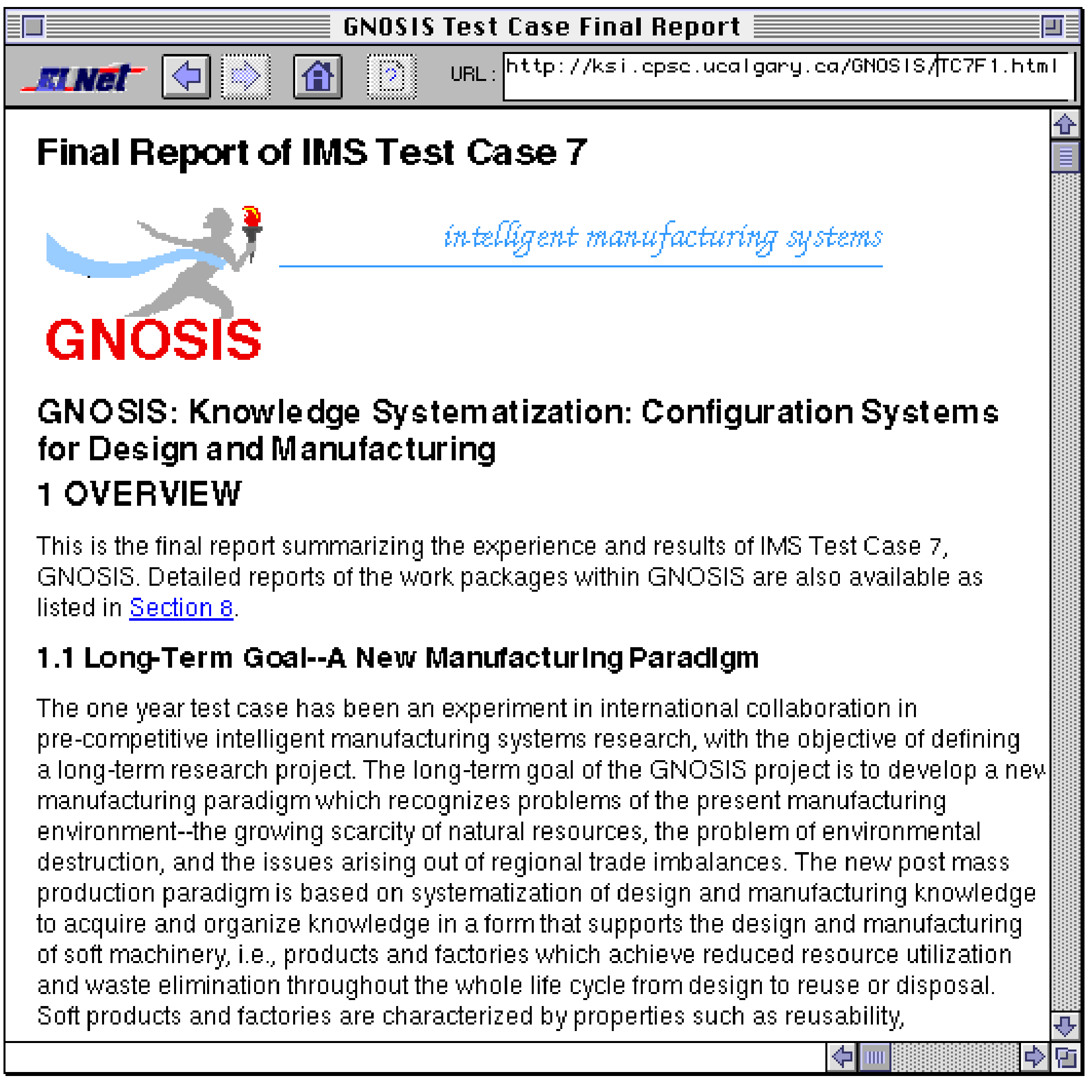


Figure 6.8 Access to hypertext multimedia document archives through World Wide Web

One can now perceive an evolutionary sequence: FTP can fetch documents across the net for viewing in other applications; Gopher can fetch and display document catalogs and simple textual documents; World Wide Web can fetch and display documents with typography, embedded images, and embedded hypertext links to other documents. Also the functionality is cumulative in that World Wide Web can FTP a document for another application and display Gopher catalogs. Figure 6.9 shows the Gopher catalog of Figure 6.7 being accessed through World Wide Web hypertext links. And World Wide Web is itself evolving to include general two-way interaction through active documents providing a client-server computing environment on the Internet (Section 9).

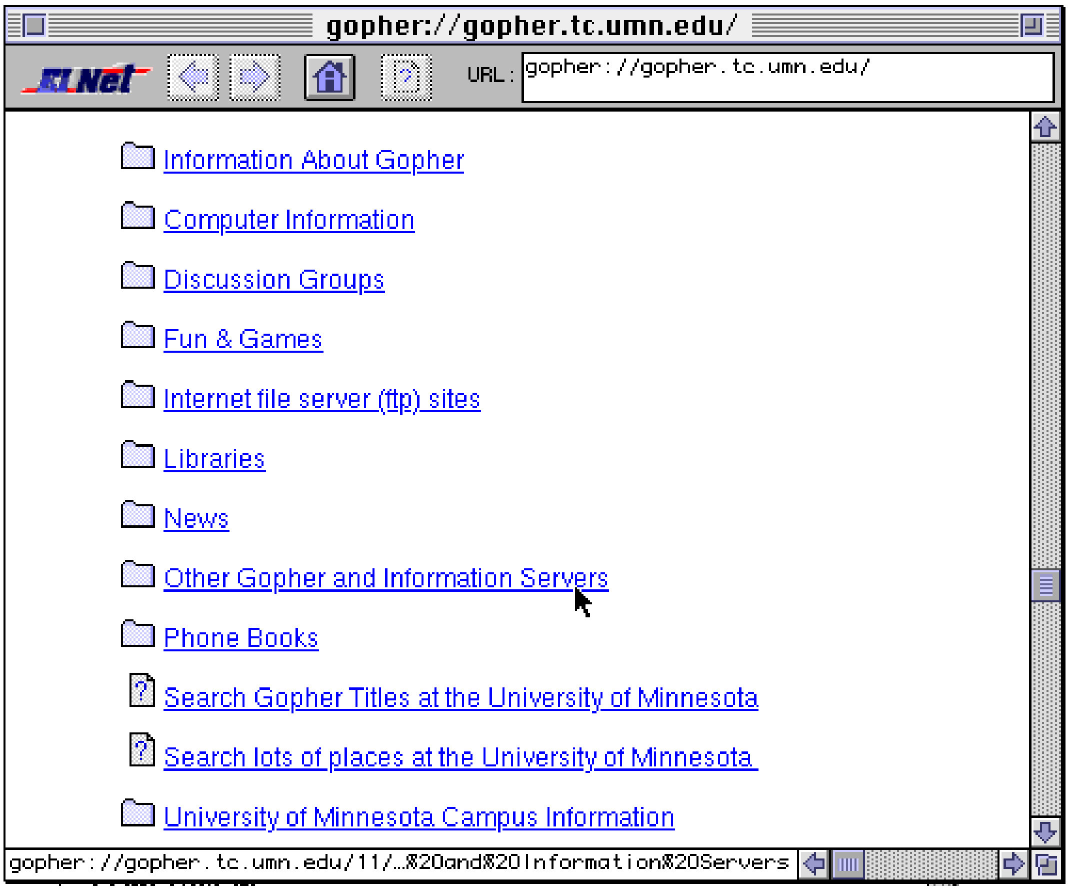


Figure 6.9 Access to Gopher through World Wide Web

6.10 Growth of Discourse on the Net

This cumulative evolution of services has led to massive growth in the use of the Internet to support access to information. Figure 6.10 shows a plot of the data transfer statistics for Gopher and World Wide Web collected by Merit NIC Services from the NSFNET backbone traffic. The exponential growth of World Wide Web usage in the past 18 months is apparent, with Gopher showing similar growth a year ago but increasing at a lower rate as the web became widely accessible. Figure 6.11 includes the FTP transfers also. These dominate in volume because they are typically used to transfer large PostScript and data files. This difference in usage makes the significance of World Wide Web growth even more apparent—it is primarily responsible for the growth of commercial and government interest in the potential of the *information highway.*

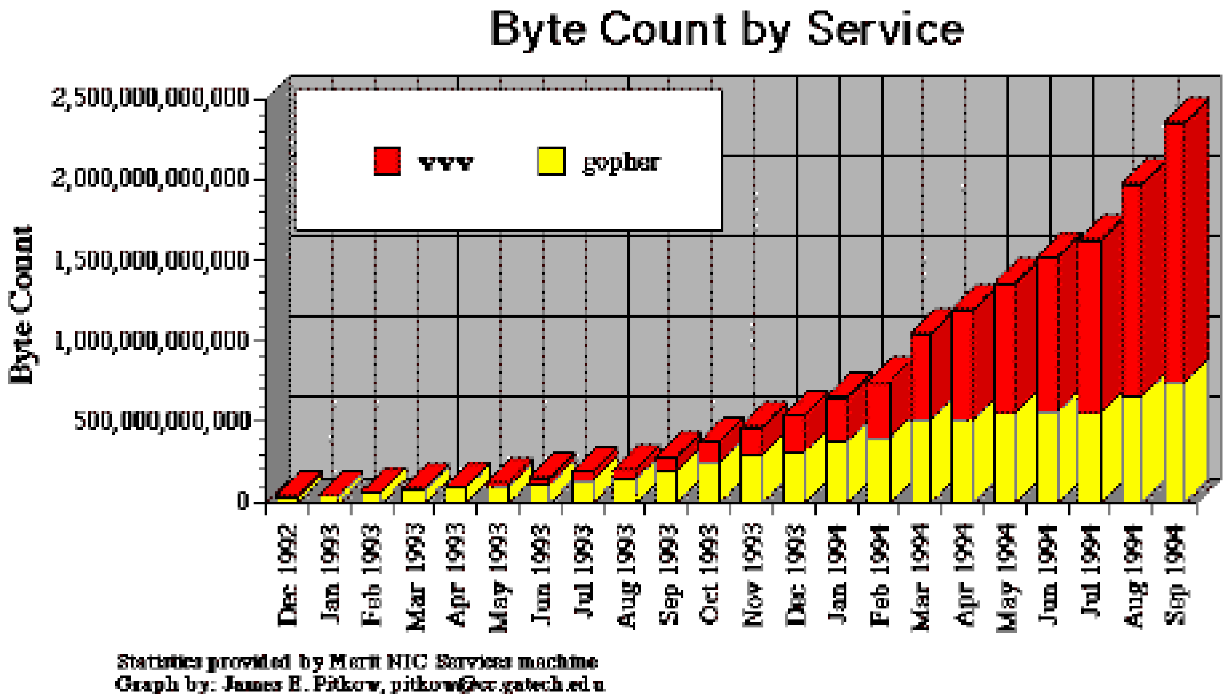


Figure 6.10 Growth of data transferred through Gopher and through World Wide Web

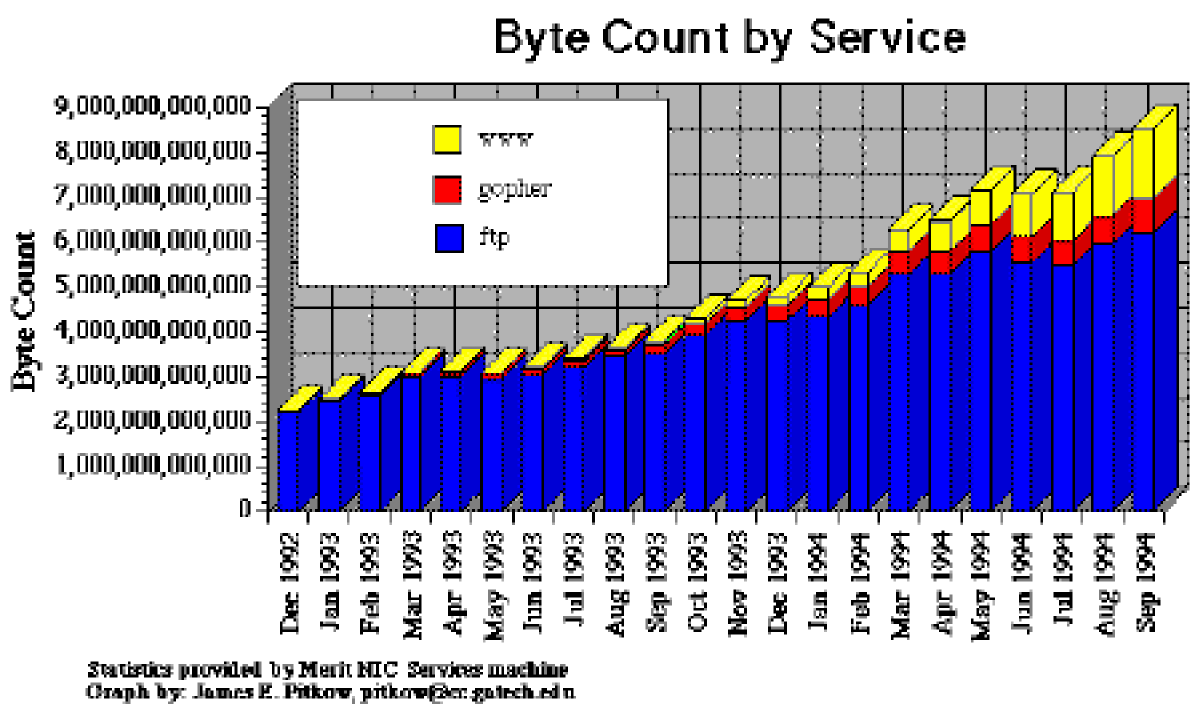


Figure 6.11 Growth of data transferred through Gopher and through World Wide Web

The statistical plots of Figures 6.10 and 6.11 were themselves obtained through World Wide Web from a server that James Pitkow has developed at the Graphics and Usability Center of Georgia Institute of Technology (http://www.cc.gatech.edu/gvu/stats/NSF/merit.htm).

7 MOVING A DOCUMENT ARCHIVE TO THE WORLD-WIDE WEB

Moving an exist document archive to the World Wide Web is a reasonably simple task that is supported by a number of public domain tools. This section explains the protocols involved and exemplifies the tools.

7.1 HTML: HyperText Markup Language

As already noted, World Wide Web documents are represented in HTML, a document type definition in the standard generalized markup language SGML. Figure 7.1 shows the HTML generating the initial section of the document of Figure 6.8. The SGML markup format is through begin and end tags, such as <html> and </html> respectively. A pair of begin and end tags encapsulates a block of text and ascribes to it the attribute noted in the tag. For example the <title> and </title> tags encapsulate the title of the document which will be displayed as the title of the window in which the document is displayed.

```
<html>
<head>
<!-- This document was created from RTF source by rtftohtml version 2.7.4 -->
<title>GNOSIS Test Case Final Report</title>
</head>
<body>
<h1><a name="RTFToC1">Final Report of IMS Test Case 7</a></h1>
<IMG SRC="TC7F11.gif">
<h2><a name="RTFToC2">GNOSIS: Knowledge Systematization: Configuration Systems for Design
and Manufacturing</a></h2>
<h2><a name="RTFToC3">1 OVERVIEW</a></h2>
<p>This is the final report summarizing the experience and results of IMS Test
Case 7, GNOSIS. Detailed reports of the work packages within GNOSIS are also
available as listed in  <a href="TC7F8.html">Section 8</a>.
<h3><a name="RTFToC4">1.1 Long-Term Goal--A New Manufacturing Paradigm</a></h3>
<p>The one year test case has been an experiment in international collaboration in
pre-competitive intelligent manufacturing systems research, with the objective
of defining a long-term research project. The long-term goal of the GNOSIS
```

Figure 7.1 HTML representation of GNOSIS World Wide Web document of Figure 6.8

The entire document is encapsulated in <html></html> tags indicating the DTD used. A header of status information about the document is encapsulated in <head></head> tags. Paragraphs are encapsulated in <p></p> tags where the </p> is generally omitted under an SGML minimization convention whereby some tags may have their presence inferred. Headings are encapsulated in <hn></hn> tags where n is one of 1 through 6 to indicate different levels of headings. Tags are defined in the HTML DTD for numbered and bulleted lists, quotations, blocks of preformatted text, and so on. They are also defined for typographic markup, either logically as in indicating emphasized words, or literally as in for text in a bold face.

The actual representation of tagged text is the responsibility of the browser which will provide some form of style sheet so that a user can define which text should be emphasized, how a heading level 1 should appear, and so on. Most users leave the style sheet set to the defaults for the browser, giving World Wide Web documents a similar appearance across machines. However, the local control of appearance can be used to address individual preferences, for example to provide larger fonts for the visually impaired.

The data for the picture of the GNOSIS logo at the top of Figure 6.8 is not embedded in the HTML representation. Instead a reference to it is given in line 8 through the tag <IMG SRC="TC7F11.gif">, and the browser fetches the image data and inserts it where the tag is placed in the text. Additional attributes may be specified in the image tag to specify its placement relative to the text if required.

The hypertext links in Figure 7.1 are particularly interesting. The   tags in line 7 encapsulate the text of the heading and give it the attribute of having the name "RTFToC2". This enables it to be used as the target of a hypertext link from another document, such as a table of contents. As already noted, the terms “Section 8” in line 14 of Figure 7.1 is a hypertext link to Section 8 of the GNOSIS Final Report, and it is encapsulated in the tags which give it the attribute of being a hypertext reference to the document TC7F8.html in the same directory.

HTML markup may appear complex if one is used to WYSIWYG word processors, but in practice it is very simple. There are many basic introductions to HTML available on the Internet. A short overview is available in the HTML Primer (HTML, 1993), and a more extensive one in a tutorial paper (Barry, 1994). The detailed technical specification and DTD are also available (Berners-Lee et al., 1994), and will shortly become an Internet RFC (nominally a “Request for Comments” but *de facto* an agreed standard). Since automatic conversion tools are available that allow documents prepared in word processors to be converted to HTML format, it is not necessary to have a deep understanding of HTML to prepare material for World Wide Web. However, it is worth browsing a primer to understand at least the basics of the document format, and the power available through references to embedded images and other material, and through hypertext links. An up-to-date guide to relevant resources on the Internet is maintained in the document *World Wide Web Frequently Asked Questions* (Boutell, 1994).

7.2 URL: Uniform Resource Locator

World Wide Web introduced a protocol for accessing a file on the Internet through a uniform resource locator specifying: the protocol to be used; the Internet address of the machine on which the file was located; the directory path to the file; the file name; and optional parameters such as a named location inside an HTML file. The syntax of a URL is:

protocol :// **address** / **path** / **file** delimiter **parameters**

where the delimiter for a named location is “#”. Thus, the URL for section 1.1 of the document of Figures 6.8 and 7.1 is:

http://ksi.cpsc.ucalgary.ca/GNOSIS/TC7F1.html#RTFToC3

indicating that the hypertext transfer protocol should be used to access it on the machine at “ksi.cpsc.ucalgary.ca” in the World Wide Web sub-directory “GNOSIS” with file name “TC7F1.html” commencing at the text named “RTFToC3”.

The gopher page in Figure 6.9 is accessed through the URL:

gopher://gopher.tc.umn.edu/

indicating that the Gopher protocol should be used to access it on the machine at “gopher.tc.umn.edu” in the default Gopher sub-directory with the default Gopher file name.

URLs are frequently embedded in documents to reference images or provide hypertext links to other documents. It is convenient, particularly if one wishes to be able to move a set of associated documents to another machine or to a CD-ROM, to be able to omit the protocol, address and path, and specify a *relative URL* in which the missing items are filled in with those for the document in which the link is embedded. For example, the tag <IMG SRC="TC7F11.gif"> at line 8 of Figure 7.1 specifies a relative URL for the file containing the image that expands to:

http://ksi.cpsc.ucalgary.ca/GNOSIS/TC7F11.gif

The URL for a file accessible through FTP specifies the protocol “ftp”, and the URL syntax provides for expansion to other protocols as they are defined.

7.3 World Wide Web Browser *Helper* Applications

The capability for an HTML document to specify a hypertext link to an arbitrary file on the Internet raises a question as to what happens when that link is followed since it is possible that file accessed will not be in a format that the browser can display. The simple, and yet very powerful answer, is that World Wide Web uses file suffixes to define file types and associated *helper* applications that can open these files. Thus, when a hypertext link is followed to a file “movie.qt” the browser will fetch the file and open it in whatever QuickTime movie viewer is specified as the browser’s helper for files with the suffix “qt”.

The capability of World Wide Web browsers to access files intended for other applications and open them in those applications means that the Web is intrinsically extensible to new types of documents and applications. One can make available files related to an application such as a computer-aided design (CAD) package, issue the user community with the CAD package, and have them modify their browsers’ preferences file to specify it as a helper. Users then have a means of accessing and exchanging the CAD files in a collaborative environment without requiring the CAD package itself to be modified in any way.

Additionally, most browsers provide the capability for another application to use an inter-application protocol to request that they fetch a file specified through its URL. Through this mechanism one can develop new applications that themselves specify hypertext access through the Internet without having to develop all the functionality of a World Wide Web browser.

7.4 HTTP: HyperText Transfer Protocol

HTML and other documents intended for use with World Wide Web are generally accessed through the HyperText Transfer Protocol (HTTP) (Berners-Lee, 1993a), a generic, stateless, object-oriented protocol that uses the Multipurpose Internet Mail Extensions (MIME) content encoding protocol (Borenstein and Freed, 1993) to transmit arbitrary data. Figure 7.2 shows the request that the browser sends to a HTTP server in order to fetch the document shown in Figure 6.8. The first line specifies a “GET” request to transmit the file with path “/GNOSIS” and name “TC7F1.html” using the HTTP protocol version 1.0. The second line specifies that the browser is “Mozilla” version 0,9 beta for the Macintosh, the name of Mosaic Corporation’s *NetScape* browser. The third line specifies that the reference originated in the document with the URL “http://ksi.cpsc.ucalgary.ca/KSI/KSI.html”. The fourth line specifies that the user is “gaines@cpsc.ucalgary.ca”, an email address obtained from the browser’s preference file. The remaining four lines specify the types of data the browser is able to accept.

```
GET /GNOSIS/TC7F1.html HTTP/1.0
User-Agent: Mozilla/0.9 beta (Macintosh)
Referer: http://ksi.cpsc.ucalgary.ca/KSI/KSI.html
From: gaines@cpsc.ucalgary.ca
Accept: *
Accept: image/gif
Accept: image/x-xbitmap
Accept: image/jpeg
```

Figure 7.2 HTTP request to fetch the document of Figure 6.8

Figure 7.3 shows the data returned by the HTTP server in response to the request. The first line specifies the time of transfer. The second line specifies that the server used is “NCSA” version 1.1. The third line specifies that the content is encoded in MIME version 1.0. The fourth line specifies that the content type is text in HTML format. The fifth line specifies the time at which the file was last modified. The sixth line specifies that the content following the next blank line is 8479 characters long. The remaining lines carry the file content as shown in Figure 7.1.

```
Date: Sunday, 30-Oct-94 00:38:21 GMT
Server: NCSA/1.1
MIME-version: 1.0
Content-type: text/html
Last-modified: Wednesday, 07-Sep-94 00:42:57 GMT
Content-length: 8479

<html>
<head>
<!-- This document was created from RTF source by rtftohtml version 2.7.4 -->
<title>GNOSIS Test Case Final Report</title>
</head>
...........................
```

Figure 7.3 HTTP reply transmitting the document of Figure 6.8

The client-server messages shown in Figures 7.2 and 7.3 are normally totally invisible to the user. However, it is helpful to know the nature of the protocol underlying the operation of the World Wide Web. The protocol is termed *stateless* because a connection is established, the request and reply are transmitted, and the connected is closed with no record being kept of the transaction (except in a log file used to monitor usage of the server). Thus, a World Wide Web user does not “log in” to the server and initiate a series of transactions, but rather each transaction is an atomic action with no continuity between them. This stateless mode of operation is appropriate to a system initially defined primarily for information retrieval. However, extensions to the HTTP and HTML protocols now allow the World Wide Web to support more advanced client-server computing and techniques have been developed to maintain state information across a sequence of transactions (Section 9).

7.5 Automatic Conversion of Word Processor Documents to HTML

The HTML tagged format is simple to understand and World Wide Web hypertext documents may be prepared in a text editor in which one enters the tags directly to produce a document like that of Figure 6.8. There are also specialist HTML text editors that make it easy to enter matched HTML tags. However, many users already use WYSIWYG word processors, and many documents for the web originate as existing word processor files. Hence, utilities have been developed to convert from word processor files to HTML documents. The basis of the conversion is usually to map the styles specified in a style sheet in the word processor to those available in HTML.

The program rtftohtml (rtf to html) is a public domain conversion utility developed by Chris Hector using a public domain decoder for Microsoft's Rich Text Format (RTF) developed by Paul Dubois. The RTF encoding scheme is able to represent every detail of complex word processing documents and is exported by the majority of mainstream word processors. In particular, it exports the style sheet information used in setting up the typographic style of a document. The converter maps the style names of the document to HTML style names using a user-defined table. It also separates pictures embedded in a document and embeds an HTML reference to them in the converted document. It creates a name for each heading in the document and creates a separate HTML contents list with hypertext links to the section names.

The process of converting a document from, say, Microsoft *Word* to HTML is to either restyle the document using a Word style sheet that corresponds to the HTML tags such as h1, bulleted text, and so on, or to define an rtftohtml table that maps the existing styles in the document to HTML tags. Then one exports the document in RTF format, opens the RTF document in rtftohtml, and the output is a corresponding HTML document, a contents list in HTML, and a set of figures in PICT format. One uses a graphic conversion utility to convert the Macintosh PICT files to the CompuServe Graphic Interchange Format (GIF) which is the most commonly used image format for World Wide Web. The resultant HTML and GIF files may then be transferred to a document archive managed by an HTTP server.

Provision is made in rtftohtml for hypertext links to be embedded in a document using special conventions involving double-underlining which is not a feature of HTML and hence can be used to indicate a link. Thus, it is possible to do all the hypertext mark up in the word processor and avoid having to edit the HTML documents created in any way. This is desirable for document maintenance. It corresponds to using a high level language for encoding the document and not interfering with the 'binary' HTML format produced by the rtftohtml 'compiler'.

The KSI has used rtftohtml to manage large World Wide Web archives created and managed entirely in Microsoft *Word*. It is a very effective approach to parallel publication in electronic/paper documents and on the web. Similar tools exists for documents in Tex, FrameMaker and other document production formats.

7.6 GIFs and Speed of Communication on World Wide Web

One of the most common errors in first establishing a World Wide Web site is to take advantage of the capability of HTML documents to contain large embedded pictures and to put up pages that include pictures that take a long time to transmit yet carry little content. There are commercial sites offering Internet 'expertise' that commence with a home page involving a number of very pretty embedded images that take so long to transmit that most users kill the transmission rather than wait to view the page!

The CompuServe GIF format uses Lempel-Ziv run-length encoding of horizontal scan lines to compress the image structure (Luse, 1993). This scheme is very effective at compressing pictures comprised of long horizontal segments of identically colored pixels, and very poor at encoding images involving pixel-by-pixel changes along such a scan line. This means that it is not the absolute size of a picture that determines the data that must be transmitted, but rather the horizontal complexity of the image. Vertically, a graded tint that changes pixel by pixel causes no problems. Thus one may include large colored images provided the horizontal color changes are few. For example, Figure 7.4 shows the Adobe Persuasion colored version of Figure 2.3 embedded in a document on the web. It is 7280 bytes long and takes about 5 seconds to transmit over a 14,400 baud modem line which is within the acceptable limits of human patience. By way of contrast the image at the top of Figure 7.5 occupies about the same area but occupies 95375 bytes and takes some 67 seconds to transmit over a 14,400 baud modem line which is far too long. However, the system shown is a university information service that is accessed only over a local area network with a data rate of over 100 Kbytes/sec so that the image is fetched in a second or so and the user sees no significant delay.

These examples show the importance of understanding the way in which the GIF encoding scheme operates, and of designing World Wide Web documents for the environment in which they are to be used. For documents to be accessed from locations having poor Internet connectivity it is best to avoid the use of images as much as possible, minimize the size of any necessary, and keep the size of the documents themselves small. On the other hand, if a system is to be used largely for local use it is possible to use large and complex images and documents with impunity.

Figure 7.6 shows the transaction times for typical data over different speed lines chosen to represent: a modem over a dial-up line; a typical Internet data rate between continents; and a typical one in North America. V32bis compression gives a data rate of about 3.4Kbytes/sec for text and 1.5Kbytes/sec for gifs. The 50 Kbytes/sec figure also typifies local ethernet communications. The main components of the transaction time are time to: perform Domain Name System (DNS) lookup; transmit the request; process the request; transmit text; transmit graphics; decode graphics; and layout text and graphics. The times shown do not take into account processing at the server because this is application-dependent, or DNS lookup since this is variable and usually cached both at the local site and within the client.

The document types have been chosen to typify a wide range of applications, and to illustrate some of the features of gif encoding and caching. The times in the first column represents a typical remote user over a modem, and it can be seen that the transmission of images is the primary factor in determining the response time. There are two factors that affect image transmission apart from communication speed. First, the run-length encoding of GIFs already discussed. Second, clients typically cache all material fetched indexed by its URL, and hence a URL that is reused comes from local storage and incurs no communication delay. The reuse of the same icon within a document and across related documents is conducive to both uniformity of style and to interactivity.

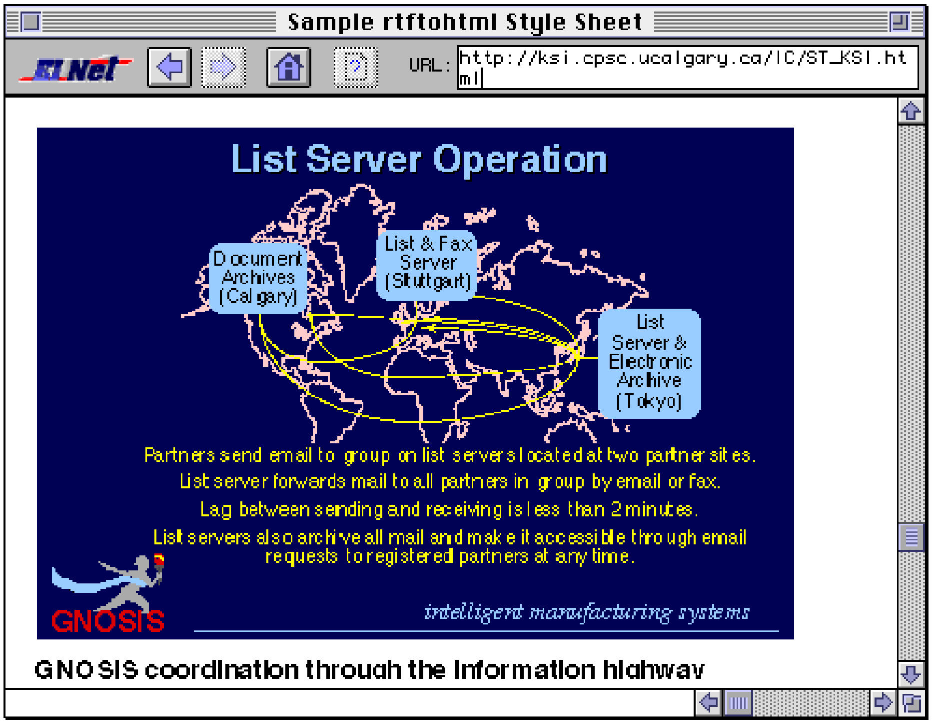


Figure 7.4 A large embedded image that compresses well and is transmitted rapidly

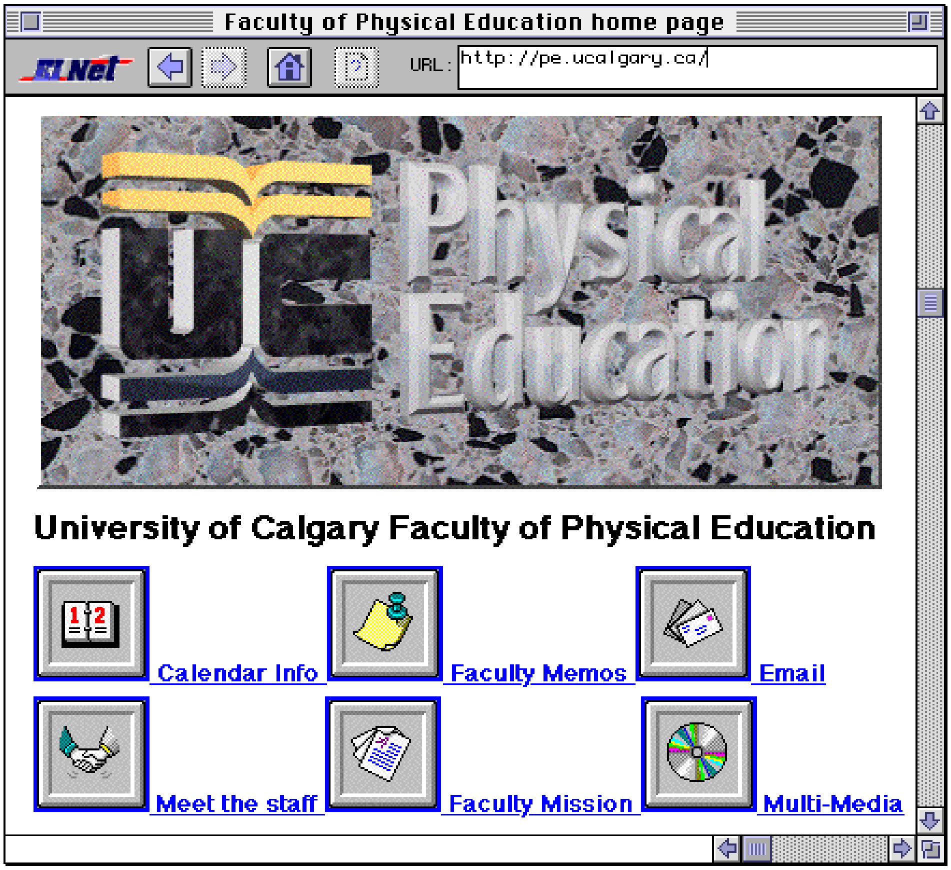


Figure 7.5 Interface to a university information system

The second and third columns show the merits of a direct connection to the Internet, and the capabilities of WWW on a local network. They also indicate that it is important to specify whether a system is intended to be used from remote sites and through modem connections.

		Transaction Time (sec)		
	Document Type text size + GIFsize **Data Rate**	V32bis modem	5 Kbytes	50 Kbytes
1	1 page paper - 2.5K+0K	1.5	1.3	0.8
2	6 pages paper - 15K+0K	5.0	3.8	1.0
3	1 page inc. 1 1/2 page fig. - 1.25K+10K	8.6	4.5	2.5
4	6 page inc. 2 1/2 page figs. - 12.5K+20K	20.6	10.3	4.4
5	6 page inc. 6 1/2 page figs. - 7.5K+60K	51.8	21.3	11.0
6	6 icon gifs (Fig. 7.5) - 0K+6x1K	13.2	10.2	9.0
7	1 page, 6 repeated bullet gifs - 2.5K+1x0.4K	3.6	3.2	2.4
8	Complex image - 0K+100K	68.1	21.5	3.5

Figure 7.6 Transaction times for different data rates

8 SETTING UP A SYSTEM TO SUPPORT A SPECIFIC COMMUNITY

Figure 8.1 shows an appropriate configuration to establish when setting up integrated support for a collaborative community on the Internet. A mailing list server handles the primary discourse of the community. If there are distinctive sub-groups or activities it may be appropriate for it to manage several mailing lists for a single community. A file transfer protocol server handles access to the document archives, supporting both uploading and downloading arbitrary files. A hypertext transfer protocol server handles access to HTML documents through World Wide Web.

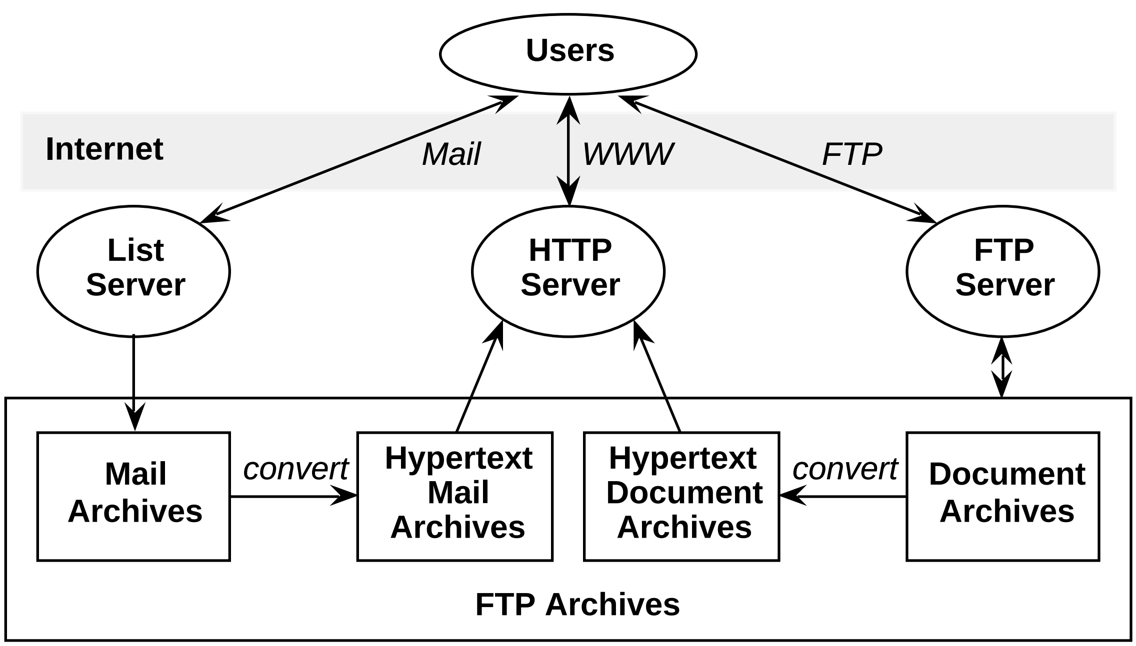


Figure 8.1 An integrated community support system on the Internet

The mail archives are made available through FTP, and they are also automatically converted to HTML format to allow them to be browsed easily through World Wide Web. Figure 8.2 shows the archives of the www-talk list server converted to HTML and indexed by topic using the conversion program *hypermail*. The FTP server gives access to all the archives, allowing new HTML documents to be uploaded to maintain the World Wide Web facilities.

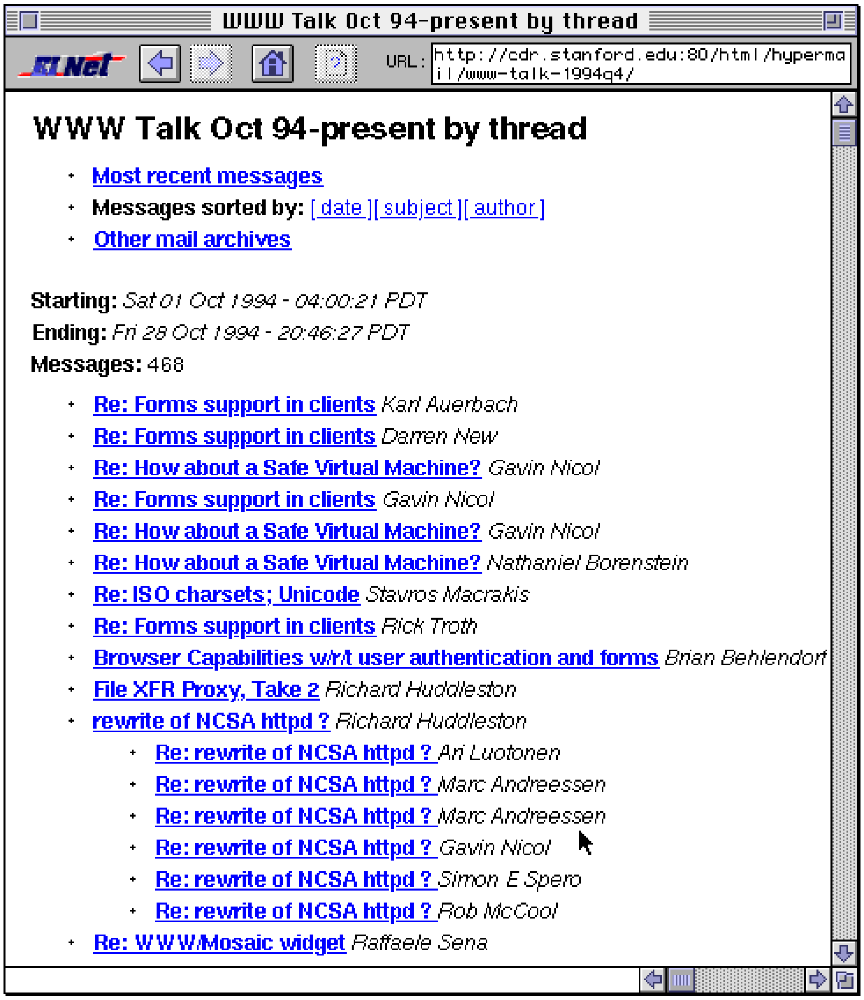


Figure 8.2 Archives of a list server converted and accessed through World Wide Web

One consideration in setting up the system shown in Figure 8.1 that has not been discussed so far is that of security. Is the system to be accessible to all those who have access to the Internet, or does the community require privacy? It is possible to set up list servers so that they are moderated and access is controlled by the moderator. It is possible to set up ftp sites so that access requires a password. It is possible to set up World Wide Web servers so that access requires authentication and documents are transmitted in an encrypted form that requires a password to decrypt. To manage security effectively requires a skilled Unix system adminstrator aware of the many problems that can arise in security on the Internet and of the techniques to overcome them. It also involves the user community in a greater effort to maintain secure passwords. Fortunately many communities are either open, or do not have major concerns about privacy, and can operate in a simple mode not requiring significant attention to security other than to ensure that archives cannot be accidentally erased or deliberately vandalized.

9 ADVANCED DEVELOPMENTS

The examples given so far have been chosen to demonstrate the use of multimedia digital formats, CD-ROM technology and Internet communications that are routinely available and simply used without computer programming. This section given an overview of related developments that show that what is already being done, no matter how significant in its own right, is only the early beginnings of the use of digital multimedia technology to support collaborative communities. Experimental systems are already in operation that are a natural evolution of World Wide Web and yet are revolutionary in the capabilities they offer. To implement such systems currently requires some degree of programming capability which the examples in the preceeding sections did not.

9.1 The Advent of HTML Forms

In March 1993 WWW was still being presented (Berners-Lee, 1993b) as primarily a hypermedia retrieval system, but in November that year a development took place that so changed the nature of WWW as to constitute a revolution in its own right. Andreessen (1993) issued NCSA Mosaic version 2 using SGML tags to encode definitions of Motif widgets embedded as *forms* within a hypermedia document, and allowed the state of those widgets within the client to be transmitted to the server. Suddenly the WWW protocols transcended their original conception to become the basis of general interactive, distributed, client-server information systems.

Figure 9.1 shows the client-server architecture of World Wide Web as it has already been described in this article. A client accesses servers on the Internet using various protocols. It communicates with various helper applications that extend its functionality. When it accesses a World Wide Web server using the HTTP protocol that server can also access various helper applications through server gateways.

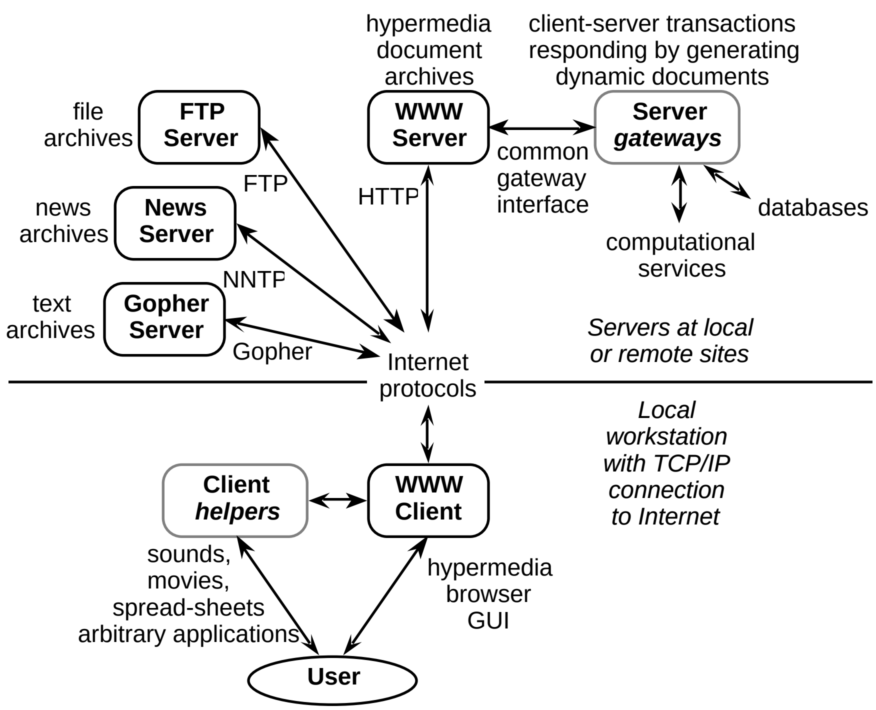


Figure 9.1 Client-server architecture of World Wide Web

What changed with the advent of the forms capability was that the client in Figure 9.1 became able to trasmit structured information from the user back to an arbitrary application gatewayed through the server. The server could then process that information and generate a HTML document which it sent back as a reply. This document could itself contain forms for further interaction with the user, thus supporting a sequence of client-server transactions.

In essence, the adoption of an SGML tagged encoding schema allows HTML documents to include not only text, typographic and multimedia material but also to carry arbitrary additional data through simple, backwards compatible extensions. The HTML DTD is currently being standardized at four levels (Berners-Lee et al., 1994):

level 0 - text with embedded links;

level 1 - adds typographic text with embedded images;

level 2 - adds embedded graphic user interfaces (forms);

level 3 - adds tables, mathematics, and…

Level 0 functionality can be supported on an alphanumeric terminal. Level 1 adds typography and pictures. Level 2 adds embedded GUIs. Level 3 is still somewhat open-ended and will evolve through prototype implementations (Raggett, 1994).

9.2 Graphic User Interfaces through HTML Forms

Figure 7.5 has already shown the genesis of using HTML documents as Graphic User Interfaces (GUIs) through hypertext links attached to pictures. It is a level 1 document resembling, and having the functionality of, an iconic GUI. To the user it appears as a computer interface where clicking on one of 6 icons gives access to one of 6 services. The document generating Figure 7.5 is shown in Figure 9.2.

```
<TITLE>Faculty of Physical Education home page</TITLE>
<IMG SRC="pelogo.gif">
<H1>University of Calgary Faculty of Physical Education</H1>
<B><A HREF="calendar.html">
<IMG SRC="cal.gif" ALT=" "> Calendar Info</A></B>
<B><A HREF="memos/readmem.html">
<IMG SRC="memo.gif"> Faculty Memos</A></B>
<B><A HREF = "execute/runmail.jcl">
<IMG SRC="mail.gif"> Email</A></B><BR>
<B><A HREF="meet.html">
<IMG SRC="meet.gif" ALT=" "> Meet the staff</A></B>
<B><A HREF="mission/mission.html">
<IMG SRC="mission.gif" ALT=" "> Faculty Mission</A></B>
<B><A HREF="media.html">
<IMG SRC="media.gif" ALT=" "> Multi-Media</A></B>
```

Figure 9.2 Document generating Figure 7.5

Clicking on an icon in Figure 7.5 requests another document but this can have the side-effect of taking other actions at either server or client. For example, the “runmail.jcl” document in line 8 of Figure 9.2 runs a script in a job control application on the client that starts the user’s normal email application and brings its window to the front.

The interface of Figure 7.5 is achieved though level 1 HTML hypermedia functionality. However, such applications suggested the provision of embedded GUIs as a natural extension to HTML. The level 2 forms extension enhances the capability of HTML documents to act as GUIs by allowing other widgets such as buttons, check boxes, radio buttons, popup menus, scrolling lists, and text entry boxes to be embedded. Figure 9.3 shows the memo entry facility selected through the “Faculty Memos” icon in Figure 7.5. The user can type information into what appears to be a normal GUI dialog box, and submit that information to the server.

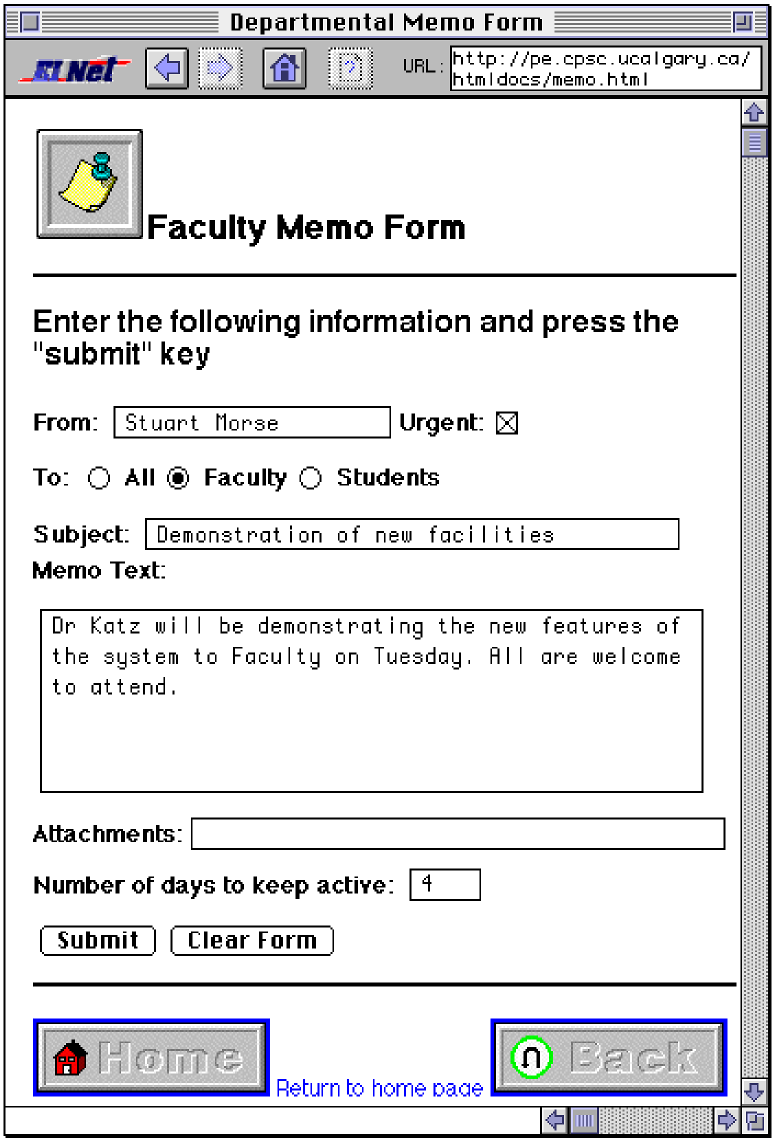


Figure 9.3 Memo entry in a university information system

Figure 9.4 shows the HTML specification for the form of Figure 9.3. The text boxes are simply specified, and the ACTION field gives the URL to which their contents will be sent using the POST facility of the HTTP protocol. When the “Submit” button is clicked the client sends the contents of the form to the server which passes them to the program “post-memo” which files them. Users may retrieve memos through an index document with hypertext links to available memos.

```
<FORM METHOD="POST" ACTION="http://pe.cpsc.ucalgary.ca/cgi-bin/post-memo">
<H2>Enter the following information and press the "submit" key</H2> <BR>
<STRONG>From: </STRONG>
<INPUT SIZE=20 NAME="username">
<STRONG>Urgent:</STRONG>
<INPUT TYPE=CHECKBOX NAME="urgent" VALUE="yes"> <BR>
<STRONG>To: </STRONG>
<INPUT TYPE=RADIO NAME="to" VALUE="All" CHECKED>
<STRONG>All</STRONG>
<INPUT TYPE=RADIO NAME="to" VALUE="Faculty">
<STRONG>Faculty</STRONG>
<INPUT TYPE=RADIO NAME="to" VALUE="Students">
<STRONG>Students</STRONG>
<BR> <STRONG>Subject: </STRONG>
<INPUT SIZE=40 NAME="subject"> <BR>
<STRONG>Memo Text: </STRONG><BR>
<TEXTAREA NAME="Content" cols=50 rows=6></textarea> <BR>
<STRONG>Attachments:</STRONG>
<INPUT SIZE=40 NAME="attachments"><BR>
<STRONG>Number of days to keep active: </STRONG>
<INPUT SIZE=4 NAME="duration"> <BR>
<INPUT TYPE=SUBMIT VALUE="Submit">
<INPUT TYPE=RESET VALUE="Clear Form">
</FORM>
```

Figure 9.4 Form generating user interface in Figure 9.3

From a human-computer interaction perspective, what is significant about Figures 7.5 and 9.3 is that they represent graphic user interfaces operating on Unix, Mac, PC and other platforms, giving access to a functionality on a local or remote server operating on any platform supporting tcp/ip connections. Moreover, the user interface is programmed through a simple script that can be written in any text editor. It is this simplicity of development of interactive applications which has generated widespread interest in programming client-server applications through the World Wide Web in the short period since the inception of the forms capability.

9.3 Group Writing through HTML Forms

The examples above were chosen because they show the capabilities of HTML forms in the rapid development of conventional GUIs. However, the embedding of the user interface in high-quality documents offers the possibility of innovative applications in which the interface functionality emerges naturally from the application context. Figure 9.5 shows a screen from GroupWriteNet (GWN), a system for supporting collaboration in document production. GWN is an extension of the KSI group writing (Gaines and Malcolm, 1993) and active document (Gaines and Shaw, 1993) tools to operate across the Internet, and part of the development of a suite of tools supporting distributed scientific communities (Gaines and Shaw, 1994c; Gaines and Shaw, 1994b).

GWN documents are stored in HTML format and can be accessed normally through WWW. A member of a registered group can request that a document belonging to that group be retrieved in annotatable form as shown in Figure 9.5. Each paragraph in the annotatable document is followed by a form allowing annotation to be entered and submitted to the server. Members of the group can then retrieve the document with each paragraph followed by hypertext links to the related annotation.

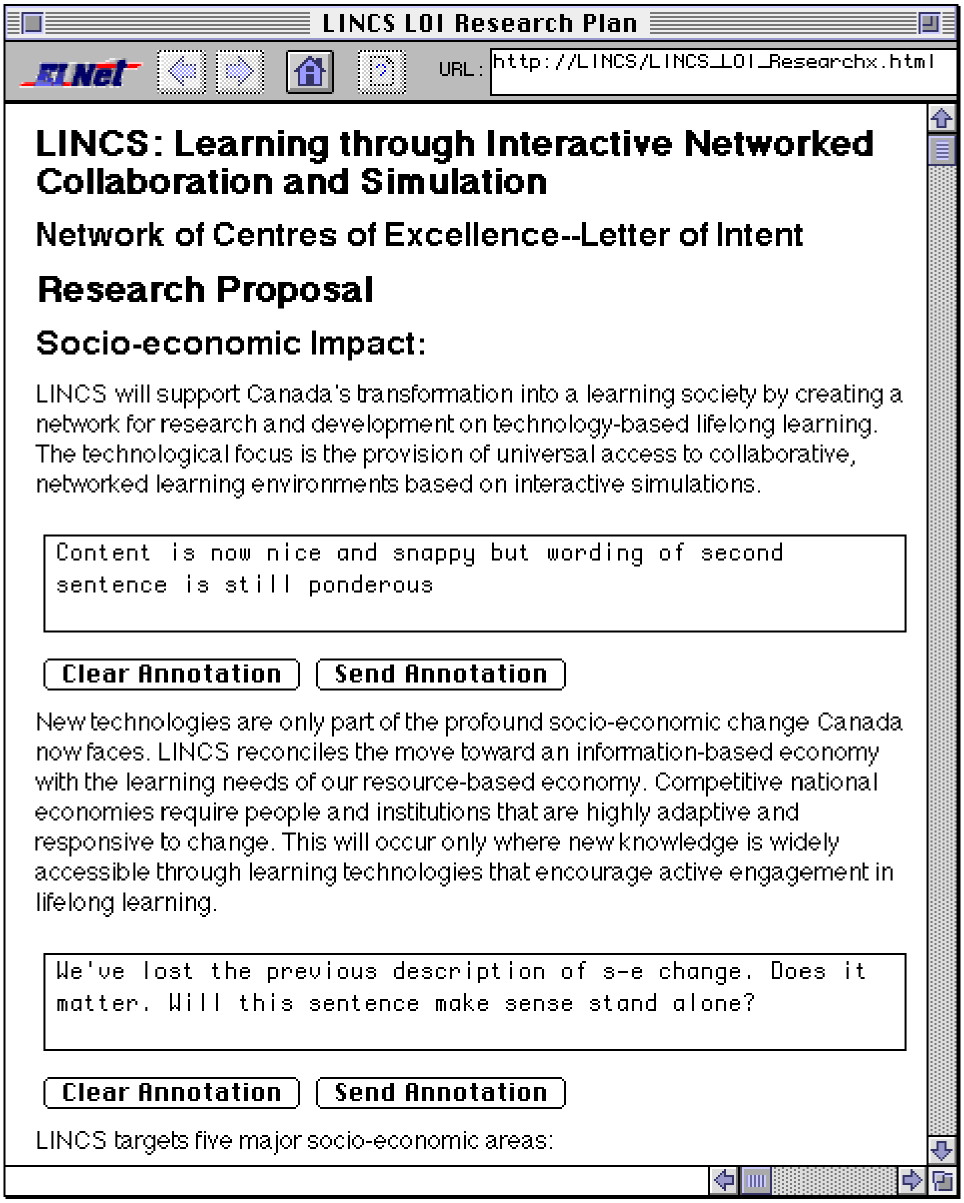


Figure 9.5 Annotation in collaborative writing

Figure 9.6 shows the HTML specification for the forms in Figure 9.5. Note the HIDDEN field in line 2 which contains an identification code for the annotator. The HTTP protocol is designed to be stateless so that the server does not in itself keep track of a sequence of related transactions with the same client. However, the required state information can be embedded in documents sent to the client so that when the server receives a request the associated service can determine the state of a transaction sequence. Since a client can save an HTML document and reopen it at a later time, this also allows transaction sequences to be suspended with the state information retained by the client. The fact that this is achieved through the normal local document storage mechanism also makes it a simple and natural activity for the user.

```
<FORM ACTION="ISC4.script" METHOD=POST>
<INPUT TYPE=HIDDEN NAME="Annotator" VALUE="BRGpGF9y">
<TEXTAREA NAME="ISC4.1.1-1" ROWS=3 COLS=60>
</TEXTAREA> <BR>
<INPUT TYPE=RESET VALUE="Clear Annotation">
<INPUT TYPE=SUBMIT VALUE="Send Annotation">
</FORM>
```

Figure 9.6 Annotation forms in Figure 9.5

9.4 Teaching Programming through HTML Forms

The level 2 HTML capabilities make WWW an excellent environment for developing client-server applications, and have triggered a wide range of innovative interactive systems. Figure 9.7 shows interaction with Ibrahim's (Ibrahim, 1994) system for teaching Pascal programming through the web. The student sees a Pascal program in a scrolling text area and can request the server to run the program, reporting the results, step through it, run to a breakpoint, and show the values of selected variables, allowing them to be changed. Breakpoints and variable to be displayed are defined by clicking on the appropriate program line.

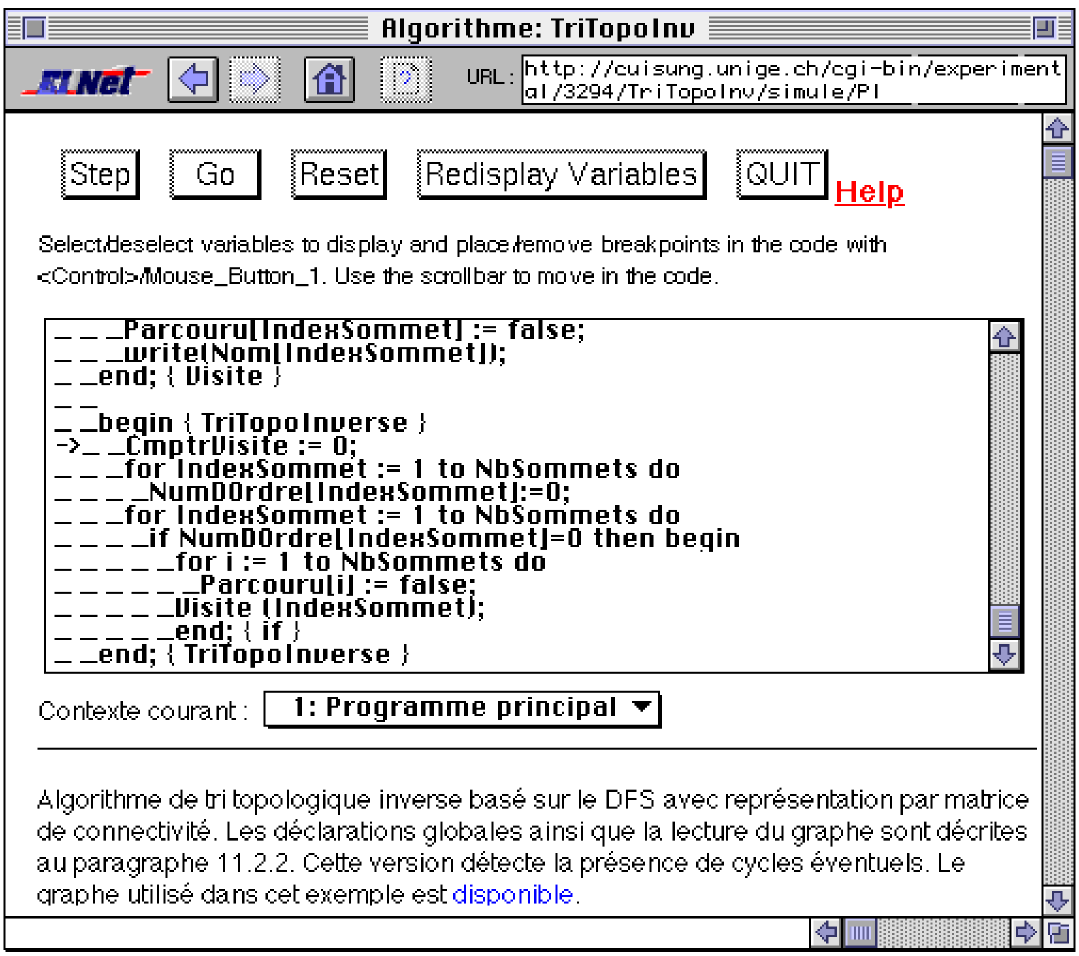


Figure 9.7 Teaching Pascal over the web

9.5 Knowledge-Based Simulation through HTML Maps

Figure 9.8 shows Gruber and Gauthier's (Gruber and Gautier, 1993) Device Modeling Environment being used in the simulation of a leak in the Space Shuttle's reaction control system.

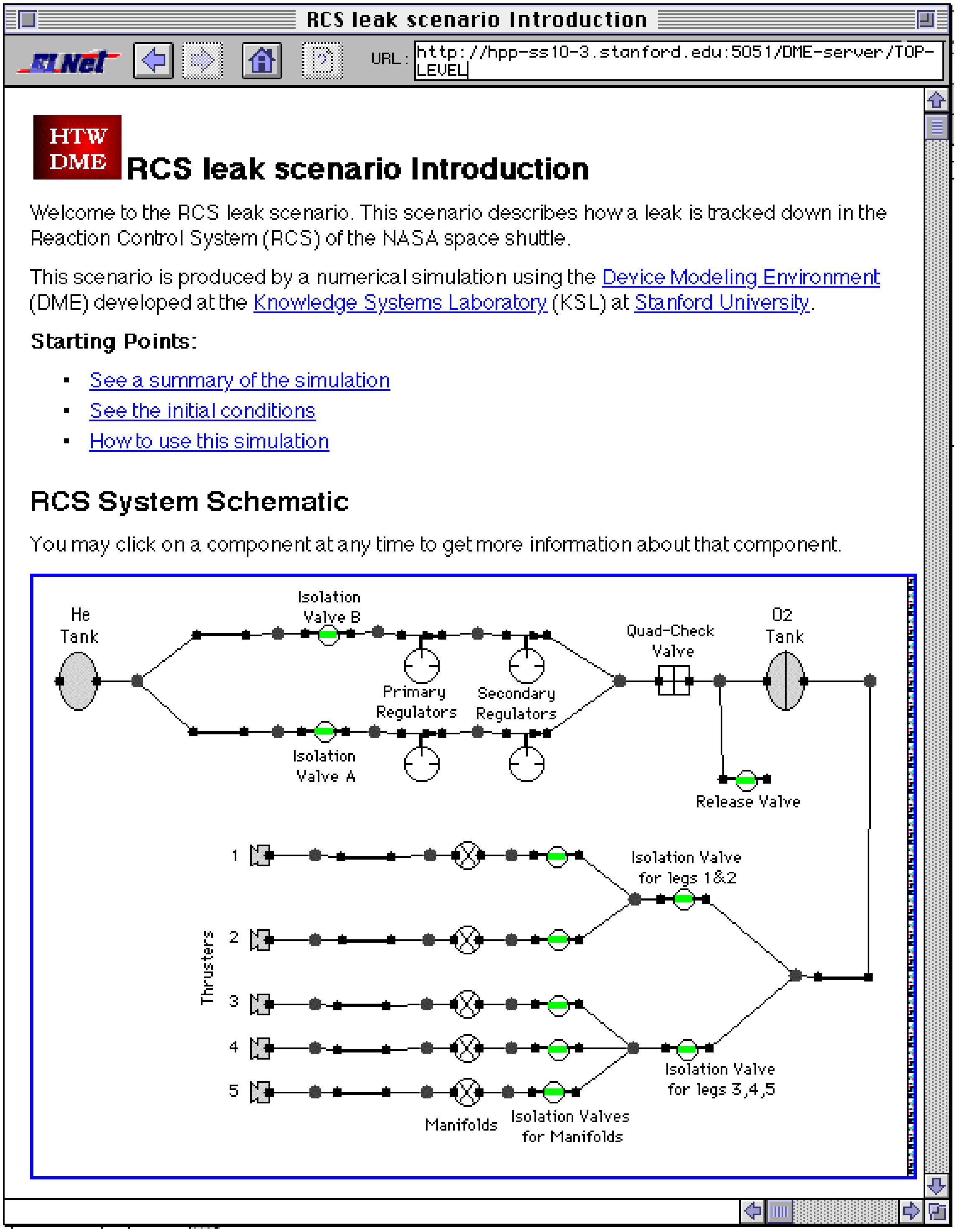


Figure 9.8 Fault diagnosis over the web

When the user clicks on a component in the schematic at the bottom of Figure 9.8 the server returns a document giving information about its role in the device model linked to other information, such as its current state. This uses another HTML level 1 interface feature, the ISMAP attribute that designates an image as a 'map' which when clicked returns the position of the cursor in the image to the server. The Device Modeling Environment system is also interesting because it uses a distributed server based on agents communicating through the kqml protocol (Finin, Weber, Wiederhold, Genesereth, Fritzson, McKay, McGuire, Shapiro and Beck, 1992) to make inferences about the model.

9.6 Concept Mapping through HTML Maps

Maps allow the web user interface to be extended with additional widgets provided a single click interaction is adequate. For example, Figure 9.9 shows the KSI's KMapServer using KMap (Gaines and Shaw, 1994a), the concept mapping tool shown indexing the GNOSIS CD-ROM in Figure 5.1, to provide access to a multimedia database. The difference between the applications is that in the CD-ROM situation KMap is running on the user's local machine whereas in the World Wide Web situation it is running on the remote server. The server uses KMap to determine in which concept the user has clicked, treating the click as if it has occurred in a server window holding the map. This allows any concept map developed in KMap to be delivered as an interactive widget on World Wide Web.

Attaching the concept mapping tool as a helper to the client allows local use of its full interactive functionality. Attaching it through a gateway to the server allows significant parts of its functionality to be offered through any client. Both are useful system architectures, and determining where to place such functionality is one of the interesting decisions to be made in designing distributed client-server applications.

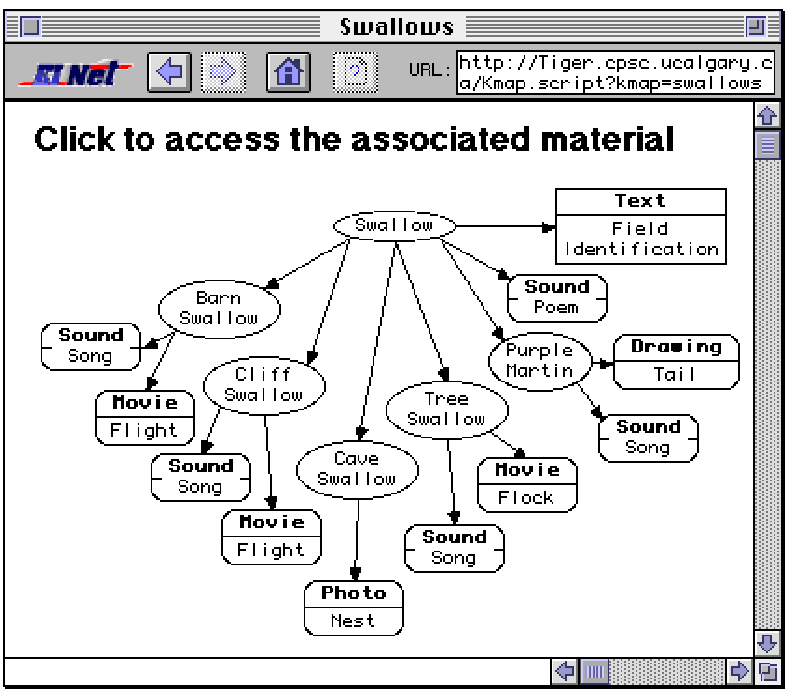


Figure 9.11 Concept map server

9.7 Implementing MUDs through HTML Forms and Maps

The Multi-User Dungeons described in Section 6.5 can also be implemented using HTML forms for user interaction, and this opens up interesting possibilities for collaborative activities on the Internet. Figure 9.12 shows the initial screen from David Blair's MUD proving access to a large archive of multimedia material derived from his film, WAX. Users can register as developers and use forms to add material to the already massive WAX environment which contains more than 900 pages of hypertext in English, French and Japanese, together with 1500 photographs, 500 video clips and 2,000 sound clips.

The multilingual, multimedia, interactive facilities of WAX illustrate what has become possible in the support of special-interest communities through MUD-like systems on World Wide Web (Rossello, 1994).

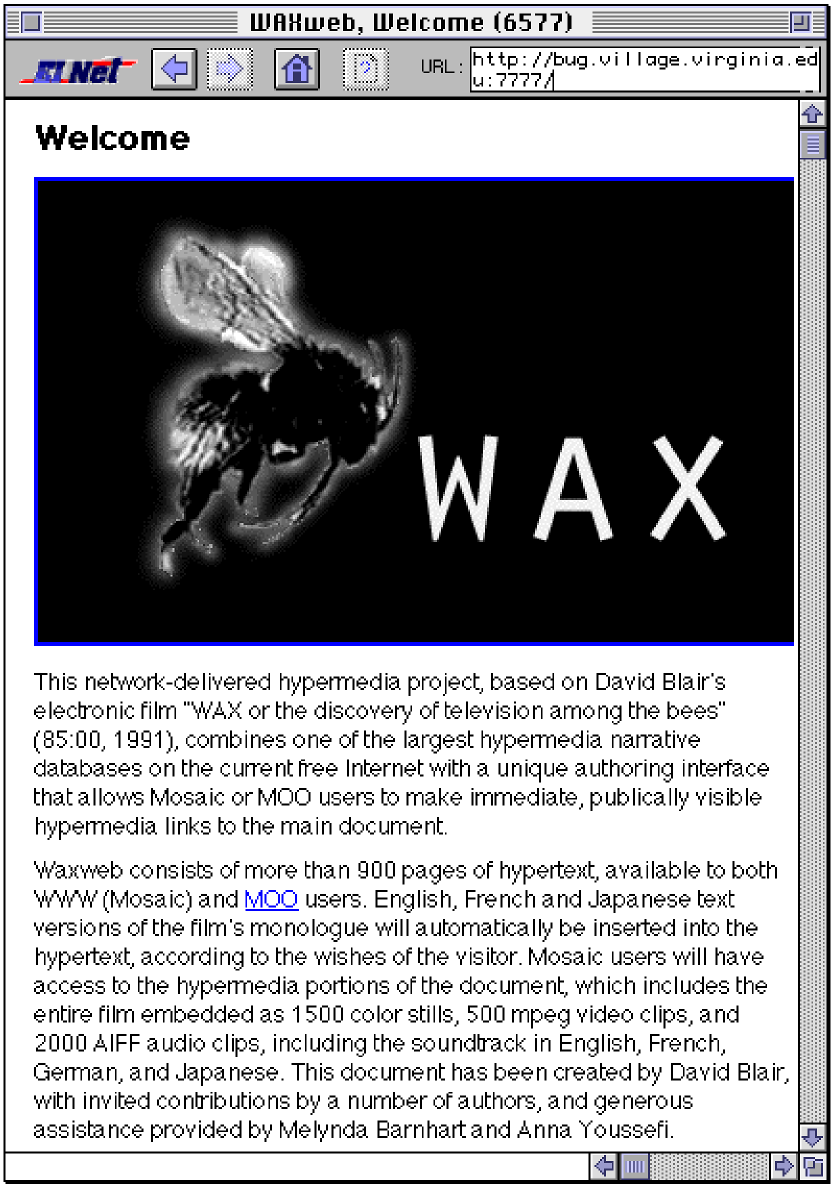


Figure 9.12 A MUD operating through World Wide Web

9.8 Telepresence through HTML Forms and Maps

Servers can interact with the real-world, for example to support tele-presence at a remote site. Figure 9.13 shows the Mercury Project (Goldberg and Mascha, 1994) developed by the Departments of Anthropology and Computer Science at USC in which users tele-operate a robot arm moving over a terrain filled with buried artifacts. A CCD camera and pneumatic nozzle mounted on the robot allow users to select viewpoints and to direct short bursts of compressed air into the terrain. Thus users can "excavate" regions within the sand by positioning the arm, delivering a burst of air, and viewing the newly cleared region. The project seeks a coherent theory that explains the buried artifacts.

The capability to support telepresecence where the user interacts with real systems in a remote situation opens up many innovative applications of World Wide Web. It changes the nature of the web from one of supporting discourse between people to one of supporting action in the real world. Experiments involving specialist equipment can take place at one site and be controlled and monitored at another. New forms of research collaboration can be explored.

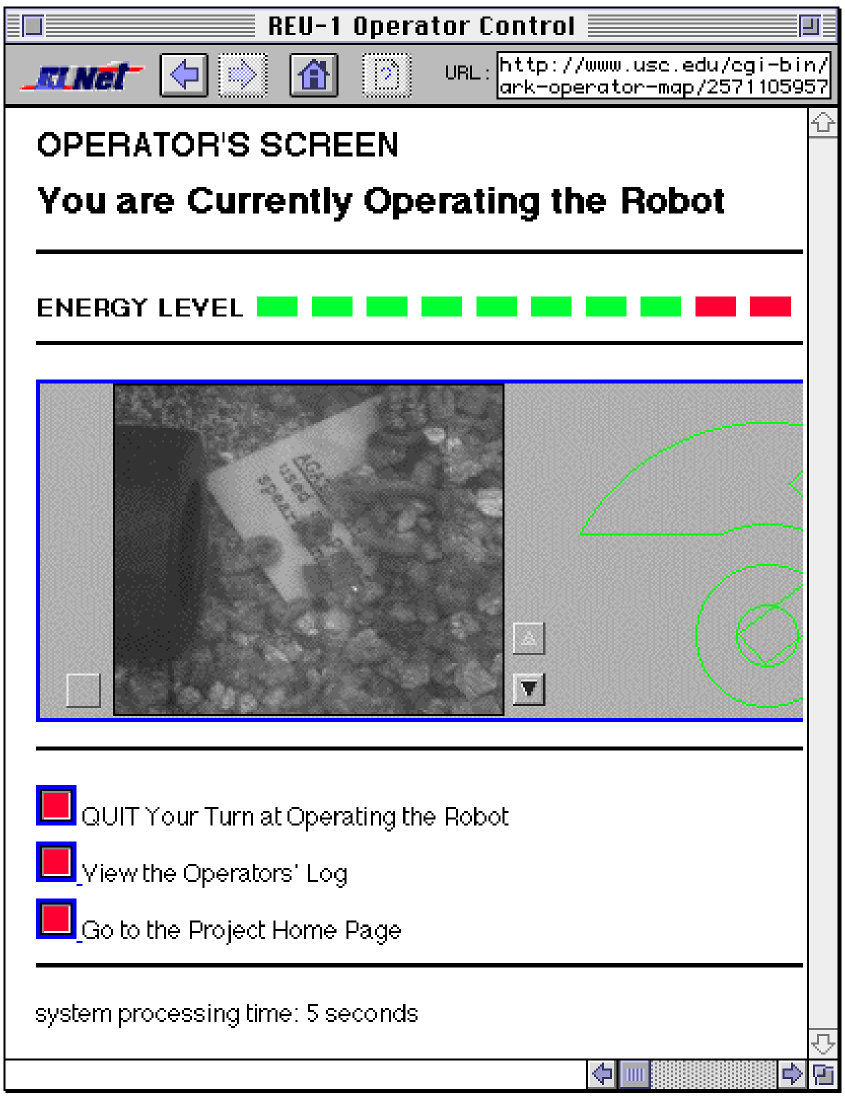


Figure 9.13 Tele-operation of a robot arm

9.9 Developments in World Wide Web Technology

World Wide Web technology is evolving rapidly. Impediments to effective interaction are being removed as they are noted. In analyzing existing limitations one must distinguish between current constraints, those that will change in the near future, and those that are comparatively long-term. For example, lack of incremental updating of the screen will disappear in a short time as clients become more intelligent, lack of down-loadable interactive widgets will disappear as a standard is developed, and delays in communication between client and server will decrease as the information highway becomes a reality. However, there may always be users with minimally-featured clients and lower speed connections, so taking such limitations into account will remain significant for many years.

The technology involves four main areas of human-computer interaction:

- **Graphic user interfaces** subject to the guidelines that have been developed over the years. The usual issues of uniformity in style of layout, presentation, vocabulary, use of color, and so on apply, and human factors design may be modeled on existing practice (Shneiderman, 1983; Smith and Mosier, 1986; Shackel and Richardson, 1991).
- **Hypermedia navigation** where issues such as cognitive overload in hyperspace navigation, and the associated guidelines, are also becoming well-defined in the hypertext community (Conklin, 1987; McKnight, Dillon and Richardson, 1991).
- **Typography, layout and visualization** which have human factors guidelines arising in graphic design and information visualization (Bertin, 1983; Tutte, 1990). The layout of HTML documents is complicated by the division of responsibilities between server and client resulting from the use of SGML.
- **Client-server computing** where communication delays and partitioning of functionality create problems that generally fall under the normal user interface guidelines. However, because they are not problems in local interaction with a personal computer, they need to be treated as peculiar to the client-server environment.

The client-server computing issues most strongly affecting the usability of WWW are:

- **Communication delays** which can be long and variable, and are crucially dependent on features of the data structures and client architecture. System designers need to be aware of far more detail about WWW operation than is apparent at first sight.
- **Limited client functionality** which interacts with delays to reduce interactivity. For example, it is difficult to give help information rapidly and in context.
- **Stateless transactions** modeled on an information retrieval paradigm. As already noted, facilities exist for the designer to track state information very effectively through hidden fields. However, the retrieval paradigm has led to the neglect of incremental updating in current clients and this again reduces interactivity.

While it is good design practice to factor out the human interface from the application functionality, communication delays on the web make it impossible to implement what has become standard practice in local application design. Designers have become used to programming the state of the interface to change as the user interacts with it so that its affordances accurately track meaningful user choices. They have also become used to providing rapid context-sensitive help that does not disturb the interaction. Neither of these is possible within the HTML level 2 specification, although the SGML tagging could allow an indefinite amount of user interface programming to be communicated to the client. Raggett (Raggett, 1994) is addressing this in the HTML level 3 specification through the provision of a script language for controlling the dynamic features of the interface.

In future developments it would be simple for clients to support script languages of the power of tcl and tk (Ousterhout, 1994), which would allow interactivity to be programmed equivalent to that of local operation. In particular this would allow widgets to be programmed offering the full variety of click and drag interfaces. There are security problems in clients supporting scripts with full access to file and operating system facilities, but these can be overcome through the use of trusted interpreters with well-defined restrictions such as those proposed for General Magic's Telescript (Knaster, 1994).

The stateless transaction protocol adds an interesting design factor to World Wide Web systems in that state information is normally associated with the computations taking place at the server rather than with the user interaction taking place at the client. However, one soon becomes used to passing state information back to a client, and making provision to store it there through operations that are natural to the user. This would be aided if clients made better provision for local action to be taken as a result of user interaction with a document. Essentially, one needs to be able to post forms to a local mini-server that can interact directly with the client. This is becoming easier as clients come to support scripting through inter-application protocols.

Another addition to client functionality that would greatly improve interactivity is the support of incremental updating of the screen. Much could be achieved by the normal flicker reduction technique of writing new data into an offscreen bitmap and refreshing the complete screen. World Wide Web was designed to retrieve a succession of different items of information, but it is being used increasingly in applications such as those illustrated in Figures 9.9 and 9.13 where the user envisions a screen of changing information not a succession of different documents.

10 CONCLUSIONS

This article was intended to be a *practice and experience* presentation complementing more sociological and strategic analyses of the future of scholarly discourse (Gaines, 1990; Gaines, 1993; Gaines, 1994b) with material explaining how to use the existing and evolving digital medium now. The level of detail necessary to understand the practical issues has made this a long article. Hopefully, the length does not detract from the utility of the paper, and the practicality is apparent of combining the technologies of multimedia digital material, CD-ROM archives, and Internet communication services. These are complementary capabilities which cumulatively support one another. The effective use of all three provides one with an eminently manageable and incredibly powerful way of supporting collaborative communities to the extent that one can seriously consider the systematic acceleration of human knowledge processes.

The final section of the article has given a glimpse of a future which is already present. It is only a year since browsers supporting the forms capability of HTML/HTTP became available. It is only six months since those browsers became available on the Macintosh and PC platforms. It is only this month that the HTML level 2.0 standard has become sufficiently widely agreed as to be issued as an Internet RFC. The Internet provides an incredibly productive environment for the rapid prototyping of systems that are instantly deployed and used by millions of people with highly creative imaginations.

It is not unreasonable to compare what we see occurring on the net with the flowering of Greek culture in the Enlightenment and with that of European culture in the Renaissance. There is a new culture on the Internet which is no longer primarily technological but instead reflects a deep and unfolding relationship between human discourse and action, and its technological support. Widespread access to the Internet means also that the culture is not geographically located, and the nature of the human-computer interface also transcends many traditional divisions based on individual characteristics. It is by no means Utopian—the net reflects humanity and is being used in a wide variety of ways that reflect both the best and the worst of human traits (Renear and Bilder, 1993; Reid, 1994).

However, it is not socio-cultural analyses that are significant in the growth in the use of multimedia digital materials, CD-ROM and Internet communications. It is the immense capabilities of these technologies to provide major resources addressing urgent needs that is the source of the explosive growth in their use. I hope this article helps potential users of these resources integrate them into their own activities effectively, and that it will serve to reduce the frustrations of harnessing attractive and powerful yet rapidly changing technologies.

ACKNOWLEDGMENTS

Financial assistance for this work has been made available by the Natural Sciences and Engineering Research Council of Canada, Industry Canada and the Isaak Walton Killam Foundation. I am grateful to Larry Katz, Stuart Morse and Mildred Shaw for access to their research materials.

REFERENCES

Anderson, A. (1991). Networks for thinking in cliques? **Science 253**(5019) 506.

Andreessen, M. (1993). NCSA Mosaic Technical Summary. NCSA, University of Illinois. ftp://ncsa.uiuc.edu/Web/Mosaic/Papers/mosaic.ps.Z.

Barry, J. (1994). The hypertext markup language (HTML) and the World Wide Web: Raising ASCII text to a new level of usability. **Public-Access Computer Systems Review 5**(5) 5-62.

Bell, D. (1973). **The Coming of Post-Industrial Society**. New York, Basic Books.

Berners-Lee, T. (1993a). Hypertext Transfer Protocol. CERN, Geneva. ftp://info.cern.ch/pub/www/doc/http-spec.ps.Z.

Berners-Lee, T. (1993b). World Wide Web Talk at Online Publishing 1993. CERN, Geneva. http://info.cern.ch/hypertext/WWW/Talks/OnlinePublishing93/Overview.html.

Berners-Lee, T. and Cailliau, R. (1990). WorldWideWeb: Proposal for a Hypertext Project. CERN, Geneva. http://info.cern.ch/hypertext/WWW/Proposal.html.

Berners-Lee, T., Connolly, D. and Muldrow, K. (1994). Hypertext Markup Language (HTML) Version 2.0. CERN, Geneva. http://www.hal.com/products/software/olias/Build-html/html-spec.ps.

Bertin, J. (1983). **Semiology of Graphics**. Madison, Wisconsin, University of Wisconsin Press.

Borenstein, N. and Freed, N. (1993). MIME (Multipurpose Internet Mail Extensions) Part One: Mechanisms for specifying and describing the format of internet message bodies. Internet. RFC 1521.

Boutell, T. (1994). World Wide Web Frequently Asked Questions. http://sunsite.unc.edu/boutell/faq/www_faq.html.

Bryan, M. (1988). **SGML: An Athor’s Guide to the Standard Generalized Markup Language**. Wokingham, UK, Addison-Wesley.

CERN (1994). History to date. CERN, Geneva. http://info.cern.ch/hypertext/WWW/History.html.

Conklin, J. (1987). Hypertext: An introduction and survey. **IEEE Computer 20**(9) 17-41.

Curtis, P. (1993). LambdaMOO Programmer's Manual. ftp://parcftp.xerox.com/pub/MOO/ProgrammersManual.ps.

Curtis, P. and Nichols, D. (1993). MUDs grow up: social virtual reality in the real world. **Proceedings Third International Conference on Cyberspace**. ftp://parcftp.xerox.com/pub/MOO/papers/MUDsGrowUp.ps.

Evard, R. (1993). Collaborative networked communication: MUDs as systems tools. **Proceedings LISA Conference**. (ftp://parcftp.xerox.com/pub/MOO/papers/Evard.ps).

Finin, T., Weber, J., Wiederhold, G., Genesereth, M., Fritzson, R., McKay, D., McGuire, J., Shapiro, S. and Beck, C. (1992). Specification of the KQML Agent-Communication Language. The DARPA Knowledge Sharing Initiative External Interfaces Working Group.

Gaines, B.R. (1985). Formation of a Knowledge Science Institute in Canada. Knowledge Science Insitute, University of Calgary. http://ksi.cpsc.ucalgary.ca/KSI/KSI_Mandate.html.

Gaines, B.R. (1990). Knowledge support systems. **Knowledge-Based Systems 3**(3) 192-203.

Gaines, B.R. (1993). An agenda for digital journals: the socio-technical infrastructure of knowledge dissemination. **Journal of Organizational Computing 3**(2) 135-193.

Gaines, B.R. (1994a). Class library implementation of an open architecture knowledge support system. **International Journal Human-Computer Studies 41**(1/2) 59-107.

Gaines, B.R. (1994b). The collective stance in modeling expertise in individuals and organizations. **International Journal of Expert Systems 7**(1) 21-51.

Gaines, B.R. and Malcolm, N. (1993). Supporting collaboration in digital journal production. **Journal of Organizational Computing 3**(2) 195-213.

Gaines, B.R. and Norrie, D.H. (1994). Mediator: information and knowledge management for the virtual factory. **SIGMAN AAAI-94 Workshop: Reasoning about the Shop Floor**. pp.30-39. Menlo Park, California, AAAI.

Gaines, B.R. and Shaw, M.L.G. (1992). Documents as expert systems. Bramer, M.A. and Milne, R.W., Ed. **Research and Development in Expert Systems IX. Proceedings of British Computer Society Expert Systems Conference**. pp.331-349. Cambridge, UK, Cambridge University Press.

Gaines, B.R. and Shaw, M.L.G. (1993). Open architecture multimedia documents. **Proceedings of ACM Multimedia 93**. pp.137-146.

Gaines, B.R. and Shaw, M.L.G. (1994a). Concept maps indexing multimedia knowledge bases. **AAAI-94 Workshop: Indexing and Reuse in Multimedia Systems**. pp.36-45. Menlo Park, California, AAAI.

Gaines, B.R. and Shaw, M.L.G. (1994b). Knowledge acquisition and representation techniques in scholarly communication. **Proceedings SIGDOC’94: ACM 12th Annual International Conference on Systems Documentation**. pp.251-260. New York, ACM.

Gaines, B.R. and Shaw, M.L.G. (1994c). Using knowledge acquisition and representation tools to support scientific communities. **AAAI’94: Proceedings of the Twelfth National**

Conference on Artificial Intelligence. pp.707-714. Menlo Park, California, AAAI Press/MIT Press.

GNOSIS (1993). TW4 Soft Machinery Workshop Proceedings. Knowledge Science Institute, University of Calgary, Canada. ftp://ksi.cpsc.ucalgary.ca/IMS/GNOSIS/TC7/tw4ws2.

GNOSIS (1994). Knowledge Systematization: Configuration Systems for Design and Manufacturing: Final Report of the Test Case. Knowledge Science Institute, University of Calgary, Canada. http://ksi.cpsc.ucalgary.ca/IMS/GNOSIS, ftp://ksi.cpsc.ucalgary.ca/IMS/GNOSIS/TC7/final.

Goldberg, K. and Mascha, M. (1994). Mercury Project. University of Southern California. http://www.usc.edu/dept/raiders/story/fmi.html.

Goldfarb, C.F. (1990). **The SGML Handbook**. Oxford, Clarendon Press.

Gruber, T. and Gautier, P. (1993). Machine-generated explanations of engineering models: a compositional modeling approach. **IJCAI'93: Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence**. pp.1502-1508 (http://www-dme.stanford.edu:5051/DME-server/DME-server/Computed/TOP-LEVEL). San Mateo, California, Morgan Kaufmann.

Habermas, J. (1981). **The Theory of Communicative Action: Reason and the Rationalization of Society**. Boston, Beacon Press.

HTML (1993). HTML Primer. NCSA. http://www.ncsa.uiuc.edu/General/Internet/WWW/HTMLPrimer.html.

Ibrahim, B. (1994). World-wide algorithm animation. **Proceedings of WWW94**. pp.http://www1.cern.ch/PapersWWW94/betrand.ps.

Knaster, B. (1994). **Presenting Magic Cap**. Reading, Massachusetts, Addison-Wesley.

Kremer, R. and Gaines, B.R. (1994). Groupware concept mapping techniques. **Proceedings SIGDOC'94: ACM 12th Annual International Conference on Systems Documentation**. pp.156-165. New York, ACM.

Krol, E. (1992). **The Whole Internet User's Guide and Catalog**. Sebastopol, California, O'Reilly.

Luse, M. (1993). **Bitmapped Graphics Programming in C++**. Reading, Massachusetts, Addison-Wesley.

Machlup, F. (1980). **Knowledge: Its Creation, Distribution and Economic Significance: Volume I: Knowledge and Knowledge Production**. Princeton, New Jersey, Princeton University Press.

Manitoba (1993). **Proceedings of the 1993 International Conference on Refereed Electronic Journals**. Winnipeg, Manitoba, University of Manitoba Library.

McKnight, C., Dillon, A. and Richardson, J. (1991). **Hypertext in Context**. Cambridge, UK, Cambridge University Press.

Ousterhout, J.K. (1994). **Tcl and the Tk Toolkit**. Reading, Massachusetts, Adison-Wesley.

Pioch, N. (1993). A Short IRC Primer. ftp://cs-ftp.bu.edu/irc/support/IRCPrimer1.1.ps.Z.

Raggett, D. (1994). A review of the HTML+ document format. **Proceedings of WWW94**. pp.http://www1.cern.ch/PapersWWW94/dsr.ps.

Reid, E.M. (1991). ELECTROPOLIS: Communication and Community on Internet relay Chat. Honours Thesis. Department of History, University of Melbourne (ftp://parcftp.xerox.com/pub/MOO/papers/electropolis.ps).

Reid, E.M. (1994). Cultural Formations in Text-Based Virtual Realities. Masters Thesis. Department of English, University of Melbourne (ftp://parcftp.xerox.com/pub/MOO/papers/CulturalFormations.ps).

Renear, A.H. and Bilder, G. (1993). Two theses about the new scholarly communication. Landow, G.P. and Delany, P., Ed. **The Digital Word: Text-based Computing in the Humanities**. pp.217-236. Cambridge, Massachusetts, MIT Press.

Rossello, R. (1994). Siggraph: virtual-reality applications and immersion experiences. **Seybold Report on Desktop Publishing 8**(12) 3-13.

Seybold (1994). Electronic document delivery. **Seybold Special Report 3**(2) 8-29.

Shackel, B. and Richardson, S., Ed. (1991). **Human Factors for Informatics Usability**. Cambridge, UK, Cambridge University Press.

Shaw, M.L.G. and Gaines, B.R. (1994). Active documents combining multimedia and expert systems. **Proceedings SIGDOC'94: ACM 12th Annual International Conference on Systems Documentation**. pp.166-175. New York, ACM.

Shneiderman, B. (1983). Direct manipulation: a step beyond programming languages. **Computer 16**(8) 57-69.

Smith, S.L. and Mosier, J.N. (1986). Guidelines for Designing User Interface Software. Mitre Corporation. ESD-TR-86-278.

Tutte, E.R. (1990). **Envisioning Information**. Cheshire, Connecticut, Graphics Press.