Network Working Group
Request for Comments: 1614
RARE Technical Report: 8
Category: Informational
C. Adie
Edinburgh University Computing Service
May 1994

Network Access to Multimedia Information

Status of this Memo

This memo provides information for the Internet community. This memo does not specify an Internet standard of any kind. Distribution of this memo is unlimited.

Abstract

This report summarises the requirements of research and academic network users for network access to multimedia information. It does this by investigating some of the projects planned or currently underway in the community. Existing information systems such as Gopher, WAIS and World-Wide Web are examined from the point of view of multimedia support, and some interesting hypermedia systems emerging from the research community are also studied. Relevant existing and developing standards in this area are discussed. The report identifies the gaps between the capabilities of currentlydeployed systems and the user requirements, and proposes further work centred on the World-Wide Web system to rectify this.

The report is in some places very detailed, so it is preceded by an extended summary, which outlines the findings of the report.

Publication History

The first edition was released on 29 June 1993. This second edition contains minor changes, corrections and updates.

Table of Contents

Acknowledgements

The following people have (knowingly or unknowingly) helped in the preparation of this report: Tim Berners-Lee, John Dyer, Aydin Edguer, Anton Eliens, Tony Gibbons, Stewart Granger, Wendy Hall, Gary Hill, Brian Marquardt, Gunnar Moan, Michael Neuman, Ari Ollikainen, David Pullinger, John Smith, Edward Vielmetti, and Jane Williams. The useful role which NCSA's XMosaic information browser tool played in assembling the information on which this report was based should also be acknowledged - many thanks to its developers.

Disclaimer

This report is based on information supplied to or obtained by Edinburgh University Computing Service (EUCS) in good faith. Neither EUCS nor RARE nor any of their staff may be held liable for any inaccuracies or omissions, or any loss or damage arising from or out of the use of this report.

The opinions expressed in this report are personal opinions of the author. They do not necessarily represent the policy either of RARE or of ECUS.

Mention of a product in this report does not constitute endorsement either by EUCS or by RARE.

Availability

This document is available in various forms (PostScript, text, Microsoft Word for Windows 2) by anonymous FTP through the following URL:

ftp://ftp.edinburgh.ac.uk/pub/mmaccess/

ftp://ftp.rare.nl/rare/pub/rtr/rtr8-rfc.../

Paper copies are available from the RARE Secretariat.

0. Extended Summary

Introduction

This report is concerned with issues in the intersection of networked information retrieval, database and multimedia technologies. It aims to establish research and academic user requirements for network access to multimedia data, to look at existing systems which offer partial solutions, and to identify what needs to be done to satisfy the most pressing requirements.

User Requirements

There are a number of reasons why multimedia data may need to be accessed remotely (as opposed to physically distributing the data, e.g., on CD-ROM). These reasons centre on the cost of physical distribution, versus the timeliness of network distribution. Of course, there is a cost associated with network distribution, but this tends to be hidden from the end user.

User requirements have been determined by studying existing and proposed projects involving networked multimedia data. It has proved convenient to divide the applications into four classes according to their requirements: multimedia database applications, academic (particularly scientific) publishing applications, cal (computeraided learning), and general multimedia information services.

Database applications typically involve large collections of monomedia (non-text) data with associated textual and numeric fields. They require a range of search and retrieval techniques.

Publishing applications require a range of media types, hyperlinking, and the capability to access the same data using different access paradigms (search, browse, hierarchical, links). Authentication and charging facilities are required.

Cal applications require sophisticated presentation and synchronisation capabilities, of the type found in existing multimedia authoring tools. Authentication and monitoring facilities are required.

General multimedia information services include on-line documentation, campus-wide information systems, and other systems which don't conveniently fall into the preceding categories. Hyperlinking is perhaps the most common requirement in this area.

The analysis of these application areas allows a number of important user requirements to be identified:

- Support for the Apple Macintosh, UNIX and PC/MS Windows environments.

- Support for a wide range of media types - text, image, graphics and application-specific media being most important, followed by video and sound.

- Support for hyperlinking, and for multiple access structures to be built on the same underlying data.

- Support for sophisticated synchronisation and presentation facilities.

- Support for a range of database searching techniques.

- Support for user annotation of information, and for user-controlled display of sequenced media.

- Adequate responsiveness - the maximum time taken to retrieve a node should not exceed 20s.

- Support for user authentication, a charging mechanism, and monitoring facilities.

- The ability to execute scripts.

- o Support for mail-based access to multimedia documents, and (where appropriate) for printing multimedia documents.

- o Powerful, easy-to-use authoring tools.

Existing Systems

The main information retrieval systems in use on the Internet are Gopher, Wais, and the World-Wide Web. All work on a client-server paradigm, and all provide some degree of support for multimedia data.

Gopher presents the user with a hierarchical arrangement of nodes which are either directories (menus), leaf nodes (documents containing text or other media types), or search nodes (allowing some set of documents to be searched using keywords, possibly using WAIS). A range of media types is supported. Extensions currently being developed for Gopher (Gopher+) provide better support for multimedia data. Gopher has a very high penetration (there are over 1000 Gopher servers on the Internet), but it does not provide hyperlinks and is inflexibly hierarchical.

Wais (Wide Area Information Server) allows users to search for documents in remote databases. Full-text indexing of the databases allows all documents containing particular (combinations of) words to be identified and retrieved. Non-text data (principally image data) can be handled, but indexing such documents is only performed on the document file name, severely limiting its usefulness. However, WAIS is ideally suited to text search applications.

World-Wide Web (WWW) is a large-scale distributed hypermedia system. The Web consists of nodes (also called documents) and links. Links are connections between documents: to follow a link, the user clicks on a highlighted word in the source document, which causes the linkedto document to be retrieved and displayed. A document can be one of a variety of media types, or it can be a search node in a similar sense to Gopher. The WWW addressing method means that WAIS and Gopher servers may also be accessed from (indeed, form part of) the Web. WWW has a smaller penetration than Gopher, but is growing faster. The Web technology is currently being revised to take better account of the needs of multimedia information.

These systems all go some way to meet the user requirements.

- o Support for multiple platforms and for a wide range of media types (through "viewer" software external to the client program) is good.

- o Only WWW has hyperlinks.

- There is little or no support for sophisticated presentation and synchronisation requirements.

- Support for database querying tends to be limited to "keyword" searches, but current developments in Gopher and WWW should make more sophisticated queries possible.

- Some clients support user annotation of documents.

- Response times for all three systems vary substantially depending on the network distance between client and server, and there is no support for isochronous data transfer.

- There is little in the way of authentication, charging and monitoring facilities, although these are planned for WWW.

- Scripting is not supported because of security issues

- WWW supports a mail responder.

- The only system sufficiently complex to warrant an authoring tool is WWW, which has editors to support its hypertext markup language.

Research

There are a number of research projects which are of significant interest.

Hyper-G is an ambitious distributed hypermedia research project at the University of Graz. It combines concepts of hypermedia, information retrieval systems and documentation systems with aspects of communication and collaboration, and computer-supported teaching and learning. Automatic generation of hyperlinks is supported, and there is a concept of generic structures which can exist in parallel with the hyperlink structure. Hyper-G is based on UNIX, and is in use as a CWIS at Graz. Gateways between Hyper-G and WWW exist.

Microcosm is a PC-based hypermedia system developed at the University of Southampton. It can be viewed as an integrating hypermedia framework - a layer on top of a range of existing applications which enables relationships between different documents to be established. Hyperlinks are maintained separately from the data. Networking support for Microcosm is currently under development, as are versions of Microcosm for the Apple Macintosh and for UNIX. Microcosm is currently being "commercialised".

AthenaMuse 2 is an ambitious distributed hypermedia authoring and presentation system under development by a university/industry consortium based at MIT. It will have good facilities for presentation and synchronisation of multimedia data, strong authoring support, and will include support for networking isochronous data. It will be a commercial product. Initial versions will support UNIX and X windows, with a PC/MS Windows version following. Apple Macintosh support has lower priority.

The "Xanadu" project is designing and building an "open, social hypermedia" distributed environment, but shows no sign of delivering anything after several years of work.

The European Commission sponsors a number of peripherally relevant projects through its Esprit and RACE research programmes. These programmes tend to be oriented towards commercial markets, and are thus not directly relevant. An exception is the Esprit IDOMENEUS project, which brings together workers in the database, information retrieval and multimedia fields. It is recommended that RARE establish a liaison with this project.

There are a variety of other academic and commercial research projects which are also of interest. None of them are as directly relevant as those outlined above.

Standards

There are a number of existing and emerging standards for structuring hypermedia applications. Of these, the most important are SGML, HyTime, MHEG, ODA, PREMO and Acrobat. All bar the last are de jure standards, while Acrobat is a commercial product which is being proposed as a de facto standard.

SGML (Standard Generalized Markup Language) is a markup language for delimiting the logical and semantic content of text documents. Because of its flexibility, it has become an important tool in hypermedia systems. HyTime is an ISO standardised infrastructure for representing integrated, open hypermedia documents, and is based on SGML. HyTime has great expressive power, but is not optimised for run-time efficiency. It is recommended that future RARE work on networked hypermedia should take account of the importance of SGML and HyTime.

MHEG (Multimedia and Hypermedia information coding Experts Group) is a draft ISO standard for representing hypermedia applications in a platform-independent form. It uses an object-oriented approach, and is optimised for run-time efficiency. Full IS status for MHEG is expected in 1994. It is recommended that RARE keep a watching brief on MHEG.

The ODA (Open Document Architecture) standard is being enhanced to incorporate multimedia and hypermedia features. However, interest in ODA is perceived to be decreasing, and it is recommended that ODA should not form a basis for further RARE work in networked hypermedia.

PREMO is a new work item in the ISO graphics standardisation community, which appears to overlap with MHEG and HyTime. It is not clear that the PREMO work, which is at a very early stage, is worthwhile in view of the existence of those standards.

Acrobat PDF is a format for representing multimedia (printable) documents in a portable, revisable form. It is based on Postscript, and is being proposed by Adobe Inc (originators of Postscript) as an industry standard. RARE should maintain awareness of this technology in view of its potential impact on multimedia information systems.

There are various standards which have relevance to the way multimedia data is accessed across the network. Many of these have been described in a previous report [1]. Two further access protocols are the proposed multimedia extensions to SQL, and the Document Filing and Retrieval protocol. Neither of these are likely to have major significance for networked multimedia information systems.

Other standards of importance include:

- MIME, a multimedia email standard which defines a range of media types and encoding methods for those types which are useful in a wider context.

- AVIs (Audio-Visual Interactive services) and the associated multimedia scripting language SMSL, which form a standardisation initiative within CCITT (now ITU-TSS) to specify interactive multimedia services which can be provided across telephone/ISDN networks.

There are two important trade associations which are involved in standardisation work. The Interactive Multimedia Association (IMA) has a Compatibility Project which is developing a specification for platform-independent interactive multimedia systems, including networking aspects. A newly-formed group, the Multimedia Communications Forum (MMCF), plans to provide input to the standards bodies. It is recommended that RARE become an Observing Member of the MMCF. A third trade association - the Multimedia Communications Community of Interest - has also just been formed.

Future Directions

Three common design approaches emerge from the variety of systems and standards analysed in this report. They can be described in terms of distinctions between different aspects of the system:

- content is distinct from hyperstructure
- media type is distinct from media encoding
- data is distinct from protocol

Distributed hypermedia systems are emerging from the research/development phase into the experimental deployment phase. However, the existing global information systems (Gopher, WAIS and WWW) are still largely limited to the use of external viewers for nontextual data. The most significant mismatches between the capabilities of currently-deployed systems and user requirements are in the areas of presentation and quality of service (i.e., responsiveness).

Improving QOS is significantly more difficult than improving presentation capabilities, but there are a number of possible ways in which this could be addressed. Improving feedback to the user, greater multi-threading of applications, pre-fetching, caching, the use of alternative "views" of a node, and the use of isochronous data streams are all avenues which are worth exploring.

In order to address these problems, it is recommended that RARE seek to adapt and enhance existing tools, rather than develop new ones.

In particular, it is recommended that RARE select the World-Wide Web to concentrate its efforts on. The reasons for this choice revolve around the flexibility of the WWW design, the availability of hyperlinks, the existing effort which is already going into multimedia support in WWW, the fact that it is an integrating solution incorporating both WAIS and Gopher support, and its high rate of growth compared to Gopher (despite Gopher's wider deployment). Gopher is the main competitor to WWW, but its inflexibly hierarchical structure and the absence of hyperlinks make it difficult to use for highly-interactive multimedia applications.

It is recommended that RARE should invite proposals for and subsequently commission work to:

- Develop conversion tools from commercial multimedia authoring packages to WWW, and accompanying authoring guidelines.

- Implement and evaluate the most promising ways of overcoming the QOS problem.

- Implement a specific user project using these tools, to validate that the facilities being developed are truly relevant to real applications.

- Use the experience gained to inform and influence the development of the WWW technology.

- Contribute to the development of PC/MS Windows and Apple Macintosh WWW clients, particularly in the multimedia data handling area.

It is noted that the rapid growth of WWW may in the future lead to problems through the implementation of multiple, uncoordinated and mutually incompatible add-on features. To guard against this trend, it may be appropriate for RARE, in coordination with CERN and other interested parties such as NCSA, to:

- Encourage the formation of a consortium to coordinate WWW technical development.

1. Introduction

1.1. Background

This study was inspired by the realisation that while some aspects of distributed multimedia technology are being actively introduced into the European research community (for instance, audiovisual conferencing, through the MICE project), other aspects are receiving less attention. In particular, one category in which there seems to be relatively little activity is providing solutions to ease remote access to multimedia resources (for instance, accessing stored audio/video clips or images, or indeed entire multimedia applications, across the network). Few commercial products address this, and the relevance of existing standards in this area is unclear.

Of the 50 or so research projects documented in the recent RARE distributed multimedia survey [1], only about six have a direct relevance to this application area. Where stated in the survey, the main research effort in these projects is often directed towards the "difficult" problems, such as the transfer of isochronous data and the design and implementation of object-oriented multimedia databases, rather than towards user-oriented issues.

This report is concerned with practical issues in the intersection of networked information retrieval, database and multimedia technologies. It aims to establish actual user requirements in this area, to look at existing systems which offer partial solutions, and to identify what additional work needs to be done to satisfy the most pressing requirements.

1.2. Terminology

In order to discuss multimedia information systems, we need a consistent terminology. The vocabulary defined below embodies some of the concepts of the Dexter hypertext reference model [2]. This model is sufficiently general to be useful for describing most of the facilities and requirements of the multimedia information systems described in this report. (However, the Dexter model does not describe searchable index objects - it is not a database reference model.)

anchor — An identified portion of a node. E.g., in a text node, an anchor might be a string of one or more adjacent characters, while in an image node it might be a rectangular area of the image.

composite node — A node containing data of multiple media types.

document — Often used loosely as a synonym for node.

hyperdocument — We refer to a collection of related nodes, linked internally with hyperlinks, as a "hyperdocument". Examples are a database of medical images and associated text; a module from a suite of teaching material; or an article in a scientific journal. A hyperdocument may contain hyperlinks to other data which exists in internally with hyperlinks, as a "hyperdocument". Examples are a other hyperdocuments, but can be viewed as largely self-contained. It is a highlevel "unit of authoring", but is not necessarily perceived as a distinct unit by a reader (although it may be so perceived, particularly if it contains few hyperlinks to outside entities).

hyperlink — Set of one or more source anchors and one or more target anchors. Also known simply as a "link".

isochronous (adjective) Describes a continuous flow of data which is required to be delivered by the network under critical time constraints.

leaf node A node which contains no source anchors.

media type An attribute of data which describes the general nature of its expected presentation. The value of this attribute could be one of the following (not exhaustive) list:

- o Text
- o Sound
- o Image (e.g., a "photograph")
- o Graphics (e.g., a "drawing")
- o Animation (i.e., moving graphics)
- o Movie (i.e., moving image)

monomedia (adjective) Said of data which is all of the same media type.

multimedia (adjective) Said of data which contains different media types. This definition is stricter than general usage, where "multimedia" is often used as a generic term for non-textual data, and where it may even be used as a noun.

physical media Magnetic or optical storage. Not to be confused with media type!

[simple] node A monomedia object which may be retrieved and displayed as a single unit.

source anchor An anchor which may be "actioned" by the user, causing the node(s) containing the target anchor(s) in the same hyperlink to be retrieved and displayed. This process is called "traversing the link".

target anchor an anchor forming part of a hyperlink, whose containing node is retrieved and displayed when the hyperlink is traversed.

2. User Requirements

User requirements in an area such as networking, which is subject to rapid technological change, are sometimes difficult to identify. To an extent, technology leads applications, and users will exploit what is possible.

2.1. Applications

Awareness of the range of networked multimedia applications which are currently being envisaged by computer users in the academic and research community leads to a better understanding of the technical requirements. This section outlines some projects which require remote access to multimedia information across research networks, and which are currently either at a preliminary stage or underway. The projects are divided into broad categories according to their characteristics.

Multimedia Databases

Here are several examples of multimedia projects which have a "database" character.

The Peirce Telecommunity Project

This project centres on the construction of a multimedia (text and image) database of the works of the American philosopher Peirce, together with tools to process the data and to make it available over the Internet. A sub-project at Brown University focuses on adapting existing client/server network tools for this purpose. The requirements for network access include facilities for structured viewing, intelligent retrieval, navigation, linking, and annotation, as well as for domainspecific processing.

Museum Object Databases

The RAMA (Remote Access to Museum Archives) project is funded under the EEC RACE II programme. Its objective is to develop a system which allows museums to make multimedia information about their exhibits and archived material available over an ISDN network. The requirements capture and technical architecture design phases are now complete, and a prototype system will be delivered in June 1993 to link the Ashmolean Museum (Oxford, GB), the Musee d'Orsay (Paris, FR) and the Museum Archeological National (Madrid, ES). Image data is the main media type of interest, although video and sound may also play a part.

The Bristol Biomedical Videodisk Project

The Bristol Biomedical Videodisc is a collection of Medical, Veterinary and Dental images. The collection holds some 24,000 still images and is continuously growing. Textual information regarding the images is included as part of the database and this can be searched on any keyword, number or other data type, or a combination of any of these. The images are currently delivered in analogue form on a videodisc, but many institutions are unable to afford the cost of videodisc players. Investigations into making this image and text database available across the network are underway.

ArchiGopher

ArchiGopher is a Gopher server at the College of Architecture, University of Michigan, dedicated to the dissemination of architectural knowledge. Presently in its infancy, ArchiGopher is intended to become a multimedia resource for all architecture faculty and students world-wide. Some of the available or planned resources are:

- The College's image bank.
- The CAD group's collection of computer models (already started).
- The Doctoral Program's recent dissertation proposals and abstracts.
- Example archive of Kandinsky paintings.
- Images of 3D CAD projects.

The principal media type in ArchiGopher is image. Files are stored in both TIFF and GIF format.

Vatican Library Exhibit

In January 1993, the US Library of Congress mounted an electronic version of the exhibition ROME REBORN: THE VATICAN LIBRARY AND RENAISSANCE CULTURE. The exhibition was subsequently processed by the University of Virginia Library. The text files were broken into individual captions associated directly with each image and a WAIS-searchable version of the object index generated. This has been made available on Gopher by the University of Virginia Library.

This project is particularly interesting, as it demonstrates some limitations of the Gopher system. The principal media types are image and text, and it is difficult to associate a caption with its image - each must be fetched separately, and using the XMosaic or xgopher client software it is not possible to tell which menu entry is the image and which the caption. (This may be a consequence of how the data has been configured for the Gopher server; if so, a requirement for better publishing tools may be indicated.) Furthermore, searching the object index will result in a Gopher menu containing references to catalogue entries for relevant exhibits, but not to the online images of the exhibits themselves, which severely limits the usefulness of the index.

It is interesting to note that during the preparation of this report, the Vatican Exhibition has been mounted on the WorldWide Web (WWW). The hypermedia presentation on the Web is very much more attractive to use than the Gopher version.

Jukebox

Jukebox is a project supported by the EEC libraries program. The project aims to evaluate a pilot service providing library users with on-line access to a database of digital sound recordings. The database will support multi-user access and use suitable storage media to make available sound recordings in a compressed format. Users will access the service with a personal computer connected to a telematic network.

Scientific Publishing

There are several refereed electronic academic journals presently distributed on the Internet. These tend to be text-only journals, and have not really addressed the issues of delivering and manipulating non-text data.

Many scientific publishers have plans for electronic publishing of existing academic journals and conference proceedings, either on physical media or on the network. The Journal of Biological Chemistry is now published on CD-ROM, for instance. Some publishers view CD-ROM as an interim step to the ultimate goal of making journals available on-line on the Internet.

The main types of non-text data which are envisaged are:

- Images. In many cases, image data (a microphotograph, say) is central to an article. Software which recognises that the text may be of secondary importance to the image is required.

- Application-specific data. The ChemLab and MoleculeLab applications are widely used, and the integration of corresponding data types with journal articles will enhance readers' ability to visualise molecular structures. Similarly, mathematics appearing in scientific papers could be represented in a form suitable for processing by applications such as Mathematica. Mathematical content could then become a much more interactive and dynamic aspect of research publications.

- Tabular data. The ability for a reader to extract tabular data from a research paper, to produce a graphical representation, to subset the data, and to further process it in a number of different ways, is viewed as an essential part of scientific electronic publishing.

- Movies. The American Astronomical Society regularly publishes videos to go with its academic journals. Electronic publishing can improve on this "hard copy" publishing by integrating video data much more closely with the source article.

- Sound. There is perhaps slightly less demand for audio information in scientific publishing, but the requirement does exist in particular specialities (such as acoustics and zoology journals).

Access to academic journals using at least four different paradigms is envisaged. Hierarchical access, perhaps using a traditional journal/volume/issue/article model, is perhaps the most obvious. Keyword searching (or full-text indexing) will be required. Browsing is another useful and often underestimated access model - to support browsing it is essential that "eye-catching" data (unlikely to be textual) is prominently accessible. The final method of access is perhaps the most important - the use of interactive viewing tools. Such tools would enable navigation of hypermedia links within and between articles, with gateways to special-purpose applications as described above. The use of these disparate access methods implies more than one structure being applied to the same underlying data.

Standards, particularly SGML, are becoming important to publishers, and it is clear that the SGML-based HyTime standard will be a front runner in providing the kind of hypermedia facilities which are being envisaged. However, progress towards a common SGML Document Type Definition (DTD) for scientific articles, even within individual publishing houses and for text-only documents, is slow.

A specific initiative involving interested parties will be required to formalise detailed requirements and to pilot standards in this area. A preliminary demonstrator project, funded by publishers and by the British Library Research and Development Department, involves making about 30 sample scientific articles available over the SuperJANET network, using a range of different software products. The demonstrator project is being managed by IOP Publishing and is being carried out at Edinburgh University Computing Service.

Existing tools, particularly WAIS and WWW, are relevant, but adequate security and charging mechanisms are required if commercial publishers are to use them. Many research groups are now making the text of preprints and published research papers available on Gopher servers.

It is interesting to note that the proceedings of the Multimedia 93 conference run by the ACM will be published electronically (on CD ROM), using a multimedia document format designed specifically for the event.

Computer-aided Learning

The ready availability of user-friendly multimedia authoring tools such as AuthorWare Professional, Asymmetrix Multimedia Toolbook, Macromind Director and many more, has stimulated much interest in multimedia for computer-aided learning applications within the user community. Sophisticated interactive multimedia courseware applications are being developed in many disparate subjects throughout the European academic community. Users are now beginning to ask network technologists, "how can I make my multimedia application available to others across the network?".

There is considerable interest in using the network to enhance delivery of multimedia teaching materials - for instance to allow students to take courses remotely (distance learning) and for their learning process to be supported, monitored and assessed remotely.

The requirements which flow from this type of network application include the ability to identify and authenticate the students using the material, to monitor their progress, and to supply on-line assessment exercises for the student to complete. Multimedia authoring tools allow very attractive presentation environments to be created, which encourages learning; this is viewed as essential by course developers. Easy-to-use authoring tools (preferably existing commercial ones) are also essential.

Finally, some learning applications involve simulations - examples include meteorological modelling and economic simulations. Network delivery of teaching materials should cope with this requirement (perhaps by acknowledging that executable scripts are just another media type).

General Information Services

There are many other possible uses of multimedia data in networked information servers which don't conveniently fall into any of the above categories. Some examples are given below.

- o On-line documentation. Manuals and instruction books often rely heavily on pictorial information, and are enhanced by dynamic media types (sound, video). The ability to access centrally-held manuals across a network makes it much easier to keep the information up-to-date.

- o Campus-wide information systems (CWIS) are an important growth area. The opportunities for enhancing such a service with multimedia data (e.g., maps) is obvious.

- o Multimedia news bulletins (e.g., the Internet Talk Radio, which is sound only).

- o Product information (the multimedia equivalent of paper advertising matter).

- o Consumer systems - e.g., tourist information servers. The utility of such systems in an academic/research environment is perhaps questionable, but it is likely that such systems will address problems which will also be met in this environment. We should be prepared to learn from such projects.

2.2. Data Characteristics

Some of the characteristics which make data more appropriate for network publication rather than publication on physical media are listed below.

- o The data may change frequently.

- o Implementing corrections and improvements to the data is very much easier.

- o It is more readily available to the data user - no purchase/delivery cycle need exist.

- o Publication on physical media may not be cost-effective for very large volumes of data. (Of course, there is a cost in networking the data as well, but the research/academic user is normally insulated from this.)

- o Access for large user communities can be established without requiring each user to purchase a potentially expensive physical media peripheral (such as a laser disk player). This is particularly helpful in classroom situations.

- o It may require less effort from the data publisher to make data available over a network, rather than set up a manual mechanism for distributing physical media.

- o If related data from many different sources is to be published, it may be more efficient to leave the data in situ, and simply publish the network addresses of the data.

There are counter-reasons which may make physical media distribution more appropriate:

- o Easier to charge for. (However, charging mechanisms do exist in some network information systems. It may be that potential information providers need to be made more aware of this.)

- o Easier to deter or prevent copyright infringement, using traditional copy-protection techniques.

2.3. Requirements Definition

From studying the applications described in the preceding section, and from discussions with the people involved with the applications, it is possible to draw up a list of general requirements which a distributed multimedia information system for the academic and research community should satisfy. These requirements are informally described in the following subsections. The descriptions are necessarily informal and incomplete: every individual application will have its own detailed requirements, which would take a great deal of effort to determine (and indeed some of the requirements may not become apparent until the application is into its development phase).

Platforms

It is clear that the European academic community, in common with other such communities, requires support for three main platforms: UNIX, Apple Macintosh, and PC/Windows. For multimedia client/server systems, the latter two are less appropriate as server platforms, but client support for all three is vital. UNIX will be most often used as the server platform.

There are other systems, such as VAX/VMS, which are also important in some sectors.

Media Types

Unsurprisingly, all applications require text data to be supported as a basic media type. Image and graphic media types are next in importance, followed by "application-specific" data (such as tabular scientific data, mathematical equations, chemical data types, etc). Sound and video media types are becoming more important as users discover how these can enhance applications.

Many different encodings are possible for each media type (e.g., image data can be encoded as TIFF, PCX, GIF, PICT and many more). An information system should not constrain the type of encoding used, and should ideally offer either a range of alternative encodings, or conversion facilities between the stored encoding and an encoding suitable for display by the client workstation.

Hyperlinks

It is clear that many applications require their users to be able to navigate through the information base according to relationships determined by the information provider - in other words, hyperlinks. Academic publishing, CAL, on-line documentation and CWIS systems all require this capability. The user should be able, by some action such as clicking on a highlighted word in a text node or on a button, to cause another node or nodes to be retrieved and displayed.

Some "hypermedia" systems are in fact simply hypertext, in that they require the source anchor of a hyperlink to be in a text node. A true hypermedia system allows hyperlinks to have their source anchors in nodes of any media type. This allows a user to click the mouse on a component of a diagram or on part of a video sequence to cause one or more related nodes to be retrieved and displayed.

Some hypermedia systems allow target anchors of a hyperlinks to be finer-grained than a whole node - e.g., the target anchor could be a word or a paragraph within a text document. Without such a capability, it is necessary for target nodes to be quite small if precision is required in a hyperlink. This may be difficult to manage, and fine-grained target anchors are therefore better.

Additional structure above or orthogonal to the underlying hyperlinked data is required in some applications. This allows the same (generally non-textual) data to be used in several different applications, or the implementation of different access paradigms.

Presentation

Related information of different media types must be capable of synchronised display. Commercial multimedia authoring packages provide many different ways of presenting, synchronising and interacting with media elements. Some of these are summarised below.

- Backdrops. An application may present all its visual information against a single background bitmap - e.g., a CAL application might use a background image of an open textbook, with graphics, text and video data all presented on the open pages of the book.

- Buttons. A "button" can be defined as an explicitly-delimited area of the display, within which a mouse click will cause an action to occur. Typically, the action will be (or can be modelled as) a hyperlink traversal. Applications use different styles of button - some may use "tabs" as in a notebook, or perhaps "bookmarks" in conjunction with the open textbook backdrop mentioned above. Others may use plain buttons in a style conforming to the conventions of the host platform, or may simply highlight a word or phrase in a text display to indicate it is "active".

- Synchronisation in space. When two or more nodes are presented together (e.g., because a link with more than one target anchor has been traversed), the author of the hyperdocument may wish to specify that they be presented in a spatially-related way. This may involve: x/y synchronisation - e.g., a video node being displayed immediately above its text caption; it may involve contextual synchronisation - e.g., an image being displayed in a specific location within a text node; or it may involve z-axis synchronisation as well - for instance a text node containing a simple title being displayed on top of an image, with the text background being transparent so that the image shows through.

- Synchronisation in time. Isochronous data may require synchronisation - the obvious case being audio and video tracks (where these are held separately). Other examples are: the synchronisation of an automatically-scrolling text panel to a video clip (for subtitling); or to an audio clip (e.g., a translation); or synchronising an animation to an explanatory audio track.

Searching

Database-type applications require varying degrees of sophistication in retrieval techniques. For applications addressed in this report, non-text nodes form the major data of interest. Such nodes have associated descriptions, which may be plain text, or may be structured into fields. Users need to be able to search the descriptions, obtain a list of "hits", and select nodes from that list to display. Searching requirements vary from simple keyword searching, via full-text indexing (with or without Boolean combinations of search words), to full SQL-style database retrieval languages.

Interaction

The user must be able to annotate documents retrieved from the information server. The annotations may be stored locally. Similarly, the user may wish to add his own (locally-held) hyperlinks to documents. (Actual modification of documents in the information system itself, or shared annotations to documents - i.e., the information system as a CSCW environment - is viewed as separate issue which this report does not address.)

If an information provider has included contact details (such as a mail address) in a document, it should be possible for the reader to invoke a program (such as a mailer) which initiates communication with the author.

In some applications, it may make sense for a user to be able to specify a region of interest in an image or movie clip, and to request a more detailed view of (or other information about) that region.

Some applications require a sequence of images to be presented under control of the user. For instance, a three-dimensional microscopic structure could be represented as a sequence of images taken with the microscope focused on a different plane for each image. For display, the user could control which image was displayed using some kind of slider control, giving the illusion of focusing a microscope. (This particular example has been taken from the Theseus project at John Moore's University, Liverpool, GB.)

Quality of Service

Research has shown [3] that user toleration of delay in computer systems depends on user perception of the nature of the requested action. If the user believes that no computation is required, tolerable delays are of the order of 0.2s. If the user believes the action he or she has requested the computer to perform is "difficult" - for instance a computation of some form - then a tolerable delay is of the order of 2s. Users tend to give up waiting for a response after about 20s. Networked multimedia information systems must be able to provide this level of responsiveness.

Management

In order to support applications involving real-money information services (e.g., academic publishing) and learning/assessment applications, there must be a reliable and secure access control mechanism. A simple password is unlikely to suffice - Kerberos authentication procedures are a possibility.

Users must be able to determine the charge for an item before retrieving it (assuming that pay-per-item will be a common paradigm alternatives such as pay-per-call, pay-per-duration are also possible). Access records must be kept by the information server for charging purposes.

Learning applications have similar requirements, except that the purpose here is not to charge for information retrieved, but to monitor and perhaps assess a student's progress.

Scripting

Many authoring packages provide scripting languages. In most cases, these languages are used to manage the presentation environment and control navigation within the hypermedia document. There are other, declarative rather than procedural, methods for achieving this, so scripting of this type is not necessarily a requirement. However, some application areas require executable scripts for other purposes (e.g., simulations in CAL applications). Care in providing such a facility is required, because of the potential for abuse (the possibility of "trojan" scripts). However, there is work going on to produce "safe" scripting languages - an example is "safe tcl", being developed by Borenstein and Ousterhout (contact ouster@cs.berkeley.edu).

Bytestream Format

For the easy transfer and handling of a hyperdocument, it must be capable of being encoded into a bytestream form, in such a way that the structure of the document is preserved and it can be decoded without loss of information.

This facility makes it possible for such documents to be supplied to a user over electronic mail, in such a way that he or she can browse them at his or her own site. This may be appropriate where the user does not have a direct connection to the Internet. It will also be useful for printing the hyperdocument.

Authoring

It is essential that a multimedia information system should have adequate authoring tools which make it easy to prepare and publish hypermedia information. Such tools need similar power to existing commercial multimedia authoring software for stand-alone multimedia applications.

3. Existing Systems

This chapter describes some existing distributed information systems in sufficient detail to reveal how they handle multimedia data, and analyses how well they meet the requirements outlined in the preceding chapter.

3.1. Gopher

The Internet Gopher is a distributed document delivery service. It allows a neophyte user to access various types of data residing on multiple hosts in a seamless fashion. This is accomplished by presenting the user with a hierarchical arrangement of nodes and by using a client-server communications model. The Gopher server accepts simple queries, and responds by sending the client a node (usually called a document in this context).

Client software is available for a large number of systems, including:

- o UNIX (character terminals)
- o X windows
- o Apple Macintosh
- o MS DOS

- o NeXT
- o VM/CMS
- o VMS
- o OS/2
- o MVS/XA

Servers are available for systems such as:

- o UNIX
- o VMS
- o Apple Macintosh
- o VM/CMS
- o MVS
- o MS DOS

Gopher was developed at the University of Minnesota.

Gopher User Image

A Gopher client offers an interface into "gopherspace", which appears to the user as a hierarchy of menus and document nodes, similar in some ways to a file system hierarchy of directories and files. Selecting an entry from a menu node causes a further menu to appear, or causes a document to be retrieved and displayed.

As well as "ordinary" document nodes, Gopher has "search nodes" when one of these is selected from a menu, the user is prompted for one or more words to search on. The result of the search is a "virtual" menu, containing entries for document nodes (within some subset of gopherspace) which match the search. A special type of Gopher search server called "veronica" provides access to a database of all directory nodes in gopherspace. This allows a user to construct a virtual menu of all Gopher menu items containing a particular word. WAIS databases may also be located at Gopher search nodes, since some Gopher servers understand the format of WAIS index files.

Gopher Protocol

Gopher uses a client-server paradigm. The Gopher protocol runs over a reliable data stream service, typically TCP, and is fully defined in RFC 1436. The following paragraphs give an overview which is sufficient for understanding how multimedia data is handled in Gopher.

A Gopher client opens a TCP connection to a Gopher server (defined by machine name and TCP port number), and sends a line of text known as the "selector" to request information from the server. The server responds with a block of data, and then closes the connection. No state is retained by the server. A null (empty) selector tells the Gopher server to return its "root" menu node, containing pointers to other information in gopherspace.

A menu is returned from a Gopher server as a sequence of lines of text, each corresponding to one entry in the menu. Each line (which is sometimes called a "Gopher reference") contains the following data, which can be used by the client software to retrieve and display the corresponding node in gopherspace.

- A single character which identifies the type of the node. Possible values of this type ID are given below.
- A human-readable string which is used by the client software when it displays the menu entry to the user.
- The selector which should be used by client software to retrieve the node. It is treated as opaque by the client software.
- The domain name of the host on which the node is held.
- The port number to use for the TCP connection.

A document node is sent by a Gopher server simply as lines of text terminated by a dot on a line by itself, or as raw binary data, with the end of the data indicated by the server closing the TCP connection. The choice depends on the type of node.

The currently-defined type IDs are as follows:

0 Node is a file.

1 Node is a directory.

2 Node is a CSO phone book server.

3 Error.

4 Node is a BinHexed Macintosh file.

5 Node is DOS binary archive of some sort.

6 Node is a UNIX uuencoded file.

7 Node is a search server.

8 Node points to a text-based telnet session.

9 Node is a binary file.

T Node points to a TN3270 connection.

Some experimental IDs are also in use:

s Node contains -law sound data.

g Node contains GIF data.

M Node contains MIME data.

h Node contains HTML data.

I Node contains image data of some kind.

i In-line text type.

The process for defining new data types and corresponding IDs is not clear.

Gopher+ Protocol

The Gopher+ protocol is an extension of the Gopher protocol. Gopher+ is defined informally in [4]. It is designed to be downwards compatible with the original protocol, so that old Gopher clients may access Gopher+ servers (without being able to take advantage of the new facilities), and Gopher+ clients may access old Gopher servers. Gopher+ is still at the experimental stage, and is liable to change.

The most important new feature is the introduction of "attributes" associated with individual nodes. The client may retrieve the attributes of a node instead of the node contents. Attributes defined so far include:

INFO — Contains the Gopher reference of the node. Mandatory.

ADMIN — Contains administrative information, including the mail address of the server administrator and the last-modified date of the node. Mandatory.

VIEWS — Contains a list of one or more "view descriptors", each of which describes an alternate view of the node. For instance, an image node may contain a TIFF view, a GIF view, a JPEG view, etc. The client software (or the user) may choose which view to retrieve. The size of the view is also (optionally) available in this attribute. The Gopher+ Attribute Registry (see below) defines the permitted view types.

ABSTRACT — This attribute contains a short description of the item. It may also include a Gopher reference to a longer abstract, held in a separate Gopher node.

ASK — This attribute is used for the interactive query extension. The interactive query facility in Gopher+ is used to obtain information from a user before retrieving the contents of a node. The client fetches the ASK attribute, which contains a list of questions for the user. His or her responses to those questions are sent along with the selector to the server, which then returns the contents of the node. This facility could be used as a very simple way of querying a database, for instance. Using the interactive query facility to supply a password for access control purposes is not a good idea - there are too many opportunities for masquerading.

The University of Minnesota maintains a registry of Gopher+ attribute types. For the VIEWS attribute, the registry contains a list of permitted view types. Note that these view types have a similar function to the type identifier described in the preceding section.

The general format of a Gopher+ view descriptor is:

```
xxx/yyy zzz: <nnnK>
```

where xxx is a general type-of-information advisory, yyy is what information format you need understand to interpret this information, zzz is a language advisory (coded using POSIX definitions), and nnn is the approximate size in bytes. Possible values for xxx include text, file, image, audio, video, terminal.

(It now appears that the University of Minnesota Gopher Team accepts the need to be consistent in the use of type/encoding attributes with the MIME specification. The Gopher+ Type Registry may thus eventually disappear, together with the set of xxx/yyy values it currently contains.)

No view descriptors for directory nodes are currently registered.

In order to make use of the information available in attributes, it is necessary to fetch the attributes before fetching the contents of a node. Gopher+ provides a way of fetching the attributes for each entry in a menu at the same time as the menu is retrieved. This saves having to establish two successive TCP connections to fetch a single document, at the expense of some additional client software complexity.

Gopher Publishing

The procedure for making data available using the Unix Gopher server "gopherd" is very straightforward. The hierarchical nature of the Unix file system closely matches the Gopher concept of menus and documents. The gopherd program exploits this - Unix directories are represented as Gopher menu nodes, and Unix files as Gopher document nodes. The names of directories and files are the entries in Gopher menus. This can lead to awkward file names containing spaces, so gopherd provides an aliasing mechanism (the \.cap directory) to get round this.

To represent menu entries pointing to Gopher nodes on other servers, special "link" files (starting with a dot) are used.

The type ID for a document node is determined from the extension of its Unix filename. If a client requests a file containing a shell script, the script is executed and the output returned to the client.

The Gopher+ version of gopherd is similar, but the .cap directory is replaced by a configuration file gopherd.conf. This file is used to specify administration attributes, and the mapping between filename extensions and view descriptors. Some limited access control (based on the client's IP address/domain name) is also provided by the Gopher+ version of gopherd.

Published Non-text Data

There is already some useful non-text data published on Gopher almost exclusively image data. See for example the Vatican Library Exhibition at the University of Virginia Library, the ArchiGopher at the University of Michigan, the weather machine at the University of Illinois. Some of these are described in the User Requirements chapter of this report.

There seem to be rather fewer sound archives in gopherspace, but interested users may access the Edinburgh University Computing Service Gopher server on gopher.ed.ac.uk, where the Testing Area contains 20 or 30 short audio files in Sun audio format. Note - the availability of this archive is not guaranteed.

Advantages

The main factor in favour of Gopher is its widespread penetration. There are over 1000 Gopher servers world-wide. This popularity is due in part to the ease of setting up a Gopher server and making information available on it, particularly on a Unix platform.

Limitations

It is unfortunate that the relatively well-defined MIME types were not adopted in Gopher+. As mentioned above, this may yet happen, although there appear to be reasons for keeping the set of MIME types small whereas Gopher requires a wide range of types to offer to clients. The latest word is that the MIME registry will be expanded to include the types which the Gopher+ developers want.

Gopher is inflexibly hierarchical in nature. Hypertext or hypermedia it is not - links to other nodes from within document nodes are not possible. There is a suggestion in the Gopher+ specification that alternate views of directory nodes could be used to provide some kind of hypermedia capability, but this does not yet exist, and it is unlikely that it could be made to work as easily as the WWW hypertext model.

There is no access control at the user level - anyone can retrieve anything on a Gopher server. There is no provision for charging for information.

3.2. Wide Area Information Server

The Wide Area Information Server (WAIS) system allows users to search for and retrieve information from databases anywhere on the Internet. WAIS uses a client-server paradigm, and client and server software is available for a wide range of platforms. Client applications are able to retrieve text or other media documents stored on the servers, by specifying keywords. The server software searches a full-text index of the documents, and returns a list of documents containing the keywords (ranked according to a heuristic algorithm). The client may then request the server to send a copy of any of the documents found. Relevant documents can be fed back to a server to refine the search. Successful searches can be automatically re-run, to alert the user when new information becomes available.

WAIS was developed by Thinking Machines Corporation of Cambridge, Massachusetts, in collaboration with Apple Computer Inc., Dow Jones and company, and KPMG Peat Marwick. The WAIS software has been made freely available; however Thinking Machines has announced that they will stop support for their publicly-distributed WAIS as of version 8b5.1. Future support and development of the publicly-distributed WAIS has been taken over by CNIDR (Clearinghouse for Networked Information Discovery and Retrieval) in the USA. Future CNIDR releases will be called FreeWAIS. A new company, WAIS Inc, has been formed by Thinking Machines to take over commercial exploitation of the Thinking Machines WAIS software.

WAIS server software is available for the following platforms:

- UNIX
- VAX/VMS

Client software is available for the following platforms:

- UNIX (versions for X, Motif, Open Look, Sun View)
- NeXT
- Macintosh
- MS DOS
- MS Windows
- VAX/VMS

There are currently over 400 WAIS databases available on the Internet. WAIS is also the basis of some commercial information services on private networks.

WAIS User Image

In order to ask a question, the user must first select one or more databases in which to look for the answer. (The list of all available databases is available from a number of well-known sites.) The next step is to enter one or more keywords as the basis of the search. The search will return a list of documents (the "result set") which contain any of the keywords. Each document is given a ranking (a number between 1 and 1000) which indicates how relevant to the user's question the server believes the document to be. The size of each document is also shown in the list. The user may limit the size of the result set - the default limit is typically 40 documents.

The user may then choose to retrieve and display one or more documents from the list. Alternatively, he or she may designate one or more documents in the list as "relevant", and perform another search to find "more documents like this". This is called "relevance feedback".

The user may retrieve general information about the database, and may examine the catalogue of all documents in the database. There is also a "database of databases", which may be searched to identify WAIS databases which may be relevant to a subject.

WAIS Protocol

The user interface (client) talks to the server using an extended version of a standard ANSI protocol called Z39.50. This is now aligned with the ISO SR (Search and Retrieval) protocol for bibliographic (library) applications, which is part of OSI. The present WAIS protocol does not utilise a full OSI stack - APDUs are transferred directly over a TCP/IP connection. The WAIS protocol is described in [5].

WAIS does not, at this time, implement the full Z39.50-1992 specification - in particular, WAIS does not permit Boolean searches (e.g., "find all documents containing 'chalk' and 'cheese' but not 'green'"). However, Boolean search capability is being added to the FreeWAIS implementation. There are facilities in the Z39.50 protocol for access control and charging, but these are not currently implemented in WAIS.

The WAIS extensions to Z39.50 are mainly to provide the relevance feedback capability.

Note that the Z39.50 protocol is not stateless - the result set may in some circumstances be retained by the server for the user to further refine or refer to. However, the subset of Z39.50 used by current WAIS implementations mean that server implementations may be stateless.

Document type is determined by the server from information in the database index (see below), and is sent to the client as part of the result set.

WAIS Publishing

The first step in preparing data for publishing in a WAIS database is to use the 'waisindex' utility. This takes a set of text files, and produces an index file which contains an occurrence list of words of three or more letters in every file. This index file is used by the WAIS server software to resolve search requests from clients.

The 'waisindex' utility indexes files in a wide range of text formats, as well as postscript and image files in various encodings (only the file name is indexed for image files). Some of the text formats involve a file as being treated as a collection of documents for the purposes of WAIS access. Note that there appears to be no formal "registry of types" - just whatever the waisindex program supports. There is no distinction between media type and encoding format.

Published Non-text Data

There is relatively little non-text data available in WAIS databases.

- o URL=wais://quake.think.com:210/CM-images is a database of TIFF images from the Connection Machine.

- o URL=wais://mpcc3.rpms.ac.uk:210/home/images/pathology/RPMS-pathology is a database of histo-pathological images and documentation on mammalian endocrine tissue.

- o URL=wais://starhawk.jpl.nasa.gov:210/pio contains GIF images from NASA planetary probe missions, together with their captions. The presence of the caption index information makes it difficult to construct a search which returns images in the result set increasing the maximum result set size may help.

Advantages

WAIS is ideally suited for its intended purpose of searching databases of textual information on the basis of keywords. It appears to have the potential to satisfy the requirements of some of the "database" category of applications mentioned in Chapter 1.

Limitations

WAIS is not (and does not pretend to be) a general-purpose information system, as Gopher and WWW are. WAIS does not have hyperlinking, and offers a purely flat structure.

A limitation which is particularly apparent is the way that the current version of FreeWAIS indexes non-text files - using only the filename! However, it does seem that simply changing the indexing program to allow a list of keywords to be attached to non-text files would suffice to allow sensible indexing of non-text data. The commercial (WAIS Inc) version of WAIS allows several files to be associated together for indexing and retrieval purposes. Furthermode, the UCSF Centre for Knowlege Management is modifying the FreeWAIS code to support the indexing of multiple content types. The document returned by WAIS will be an HTML document containing pointers to the multimedia data. Contact dcmartin@library.ucsf.edu for further information.

WAIS is not a fully-featured query/response protocol such as SQL. It has no concept of fields, or numeric data types.

It appears to be impossible to retrieve a document from its catalogue entry in many of the existing databases.

3.3. World-Wide Web

The World-Wide Web project (also known as WWW or W3), started and driven by CERN, is a large-scale distributed hypertext system. It uses the standard client-server paradigm, with client "browser" software responsible for fetching and displaying data. Originally aimed at the High Energy Physics community, it has spread to other areas.

Browser software is available for a large number of systems including:

- Line-mode dumb terminal.
- Terminal with Curses support
- Macintosh
- X/Motif
- X11
- PC/MS Windows

- o NeXT

There is server software available for:

- o VM mainframes.
- o UNIX
- o Macintosh
- o VMS

WWW User Image

The WWW world consists of nodes (usually called documents) and links. Links are connections between documents: to follow a link, a reader clicks with a mouse on a word in the source document, which causes the linked-to document to be retrieved and displayed. (On systems without a mouse, the user types a number instead.)

Indexes are special documents which, rather than being read, may be searched. To search an index, a reader supplies keywords (or other search criteria). The result of a search is a "virtual" document containing links to the documents found. All documents, whether real, virtual or indexes, look similar to the reader.

The WWW addressing mechanism means that an interface to Gopher and anonymous FTP information sources may be established, in a way which is transparent to the user. Thus, the whole of gopherspace is part of the Web. Transparent gateways to other systems, including Hyper-G and WAIS, are also available.

URL

All nodes on the Web are addressed using the "Universal [or Uniform] Resource Locator" (URL) syntax, defined in [6]. This is an Internet Draft produced by the IETF URL Working Group.

A URL is a name for an object (which may be a document or an index) on the Internet. It has the general form:

```
<scheme> : <path> [ # <anchorid> ]
```

The <scheme> identifies an access protocol or method for the object. Some of the schemes are HTTP (the native WWW protocol), anonymous FTP, Andrew file system, news, WAIS, Gopher. The <path> component locates the document in a way significant for the access method.

Thus for instance for anonymous FTP, the path includes the fully qualified domain name of the host on which the document resides, and the directory and file name under which it may be found. For some schemes, the <path> may include a search string (or combination of strings) which is used to address a "virtual" object formed by searching an index of some kind. The HTTP, WAIS and Gopher schemes can use search strings, which usually follow the rest of the path, separated from it by a ?.

The optional <anchorid> is used for addressing within an object. Its interpretation is not defined in the URL specification.

"Partial" URLs may be specified. These are used within a document on the Web to refer to another "nearby" document - for instance to a document in another file on the same machine. Certain parts of the URL (e.g., the scheme and machine name) may be omitted, according to well-defined rules. This makes it much easier to move groups of documents around, while maintaining the links within and between them.

A URL locates one and only one object on the Internet. However, more than one URL may point to the same object. Given two URLs, it is not in general possible to determine whether they refer to the same object. Furthermore, there is no guarantee that a single URL will refer to the same object at different times (the object may change incrementally, or it may be completely replaced with something different, or it may indeed be removed).

HTTP

HTTP (HyperText Transfer Protocol) is the protocol employed between server and client. It is defined in [7]. The protocol is currently being revised (see the Future Developments section below), and will eventually be proposed as an Internet standard.

The original protocol is extremely simple, and requires only a reliable connection-oriented transport service, typically TCP/IP.

The client establishes a connection with the server, and sends a request containing the word GET, a space, and the partial URL of the node to be retrieved, terminated by CR LF. The server responds with the node contents, comprising a text document in the Hypertext Markup Language (HTML). The end of the contents is indicated by the server closing the connection.

HTML

HTML (HyperText Markup Language) is the way in which text documents must be structured if they are to contain links to other documents. Non-HTML text documents may of course be made available on the Web, but they may not contain links to other documents (i.e., they are leaf nodes), and they will be displayed by browsers without formatting, probably using a fixed-width font. Like HTTP, HTML is also undergoing enhancement, but the original version is defined in [7], and is being submitted as an Internet draft.

HTML is an application of SGML (Standard Generalized Markup Language). It defines a range of useful tags for indicating a node title, paragraph boundaries, headings of several different levels, highlighting, lists, etc. Anchors are represented using an <A> tag.

For instance, here is an example of HTML containing an anchor:

```
The HTTP protocol implements the WWW <A NAME=13
HREF="../../Administration/DataModel.html">data model</A> .
```

The location of the anchor is the text "data model". It is a source anchor, with a target given by the URL in the HREF attribute, so the text would appear highlighted in some way in a client's window, to indicate that clicking on it would cause a hyperlink to be traversed. It is also a target anchor, with an anchor ID given by the NAME attribute. A source anchor referring to this target would specify #13 at the end of the node's URL. Traversing a hyperlink to this node would cause the entire node to be retrieved, but the target anchor text would be displayed in some emphasised way - for instance if the retrieved text is displayed in a scrolling window, it might be positioned such that the target anchor appears at the top of the window.

Another attribute of the <A> element, TYPE, is also available, which is intended to describe the nature of the relationship modelled by the link. However, this is not in extensive use, and there appears to be no registry of the possible values of such types.

Future Developments

HTTP and HTML are currently being extended in a backward-compatible way to add multimedia facilities. [8] describes the HTTP2 protocol. The revised HTML is defined in [9]. Both documents are subject to change (and indeed the HTML2 specification has changed substantially during the preparation of this report).

The revised HTML contains many enhancements which are useful for multimedia support. Some of the most relevant are listed below.

- o "Universal Resource Numbers" are a proposed system for unique, timeless identifiers of network-accessible files presently being designed by IETF Working Groups. URNs must be distinguished from URLs, which contain information sufficient to locate the document. URNs may be allocated to nodes and may be represented in source anchors. This saves client software from retrieving a copy of something it already has - allowing sensible caching of large video clips, for instance. The disadvantage is that when something is changed and given a new URN, the source anchors of all links which point to it must be changed (and the URNs of these documents must therefore be changed, and so on). Therefore, it makes sense to allocate URNs only to very large documents which change rarely, and not to the documents which reference them.

- o The title of a destination document may be included as anattribute of a source anchor. This allows a client to display the title to the user before or during retrieval, and also allows data which does not itself contain a title (e.g., image data) to be given one.

- o There is provision for in-line non-text data (e.g., images, video, graphics, mathematical equations), which appears in the samewindow as the main textual material in the node.

- o The concept of the relationship expressed by a hyperlink is expanded. Both source and target anchors may contain relation attributes which point forwards and backwards respectively. Possible relationships include "is an index for", "is a glossary for", "annotates", "is a reply to", "is embedded in", "is presented with". The last two are useful for multimedia - for instance, the "embed" relationship could cause a retrieved image to be fetched and embedded in the display of a text node, and the "present" relationship could cause a sound clip to be automatically retrieved and presented along with a text node.

The HTTP2 protocol maintains the same stateless connect/request/response/close procedure as the current HTTP protocol. Data is transferred in MIME-shaped messages, allowing all MIME data formats (including HTML) to be used. As well as the GET operation, HTTP2 has operations such as:

HEAD Fetch attribute information about a node (including the media type and encoding)

CHECKOUT/CHECKIN/PUT/POST

These allow nodes to be checked out for updating and checked back in again, and new nodes to be created. New node data is supplied in MIME shape with the request.

The request from the client can contain a list of formats which the client is prepared to accept, user identification, authorisation information (a placeholder at present), an account name to charge any costs to, and identification of the source anchor of the hyperlink through which the node was accessed.

The response from the server may contain a range of useful attributes (e.g., date, cost, length - but only for non-text data). The server may redirect the query, indicating a new URL to use instead. It may also refuse the request because of authorisation failure or absence of a charge account in the request.

The protocol also contains a mechanism which is designed to allow the server to make an intelligent decision about the most appropriate format in which to return data, based on information supplied in the request by the client. This may for instance allow a powerful server to store the uncompressed bitmap of an image, but to compress it on request using an appropriate encoding, according to the decoding capabilities announced by the client.

An HTTP2 server and client are currently under test. Some HTML2 features are already fitted to the XMosaic browser.

Mosaic

The Mosaic project, located at the US National Centre for Supercomputing Applications (NCSA) at the University of Illinois, is developing a networked information system intended for wide-area distributed asynchronous collaboration and hypermedia-based information discovery and retrieval. Mosaic, which is specifically oriented towards scientific research workers, has adopted the World Wide Web as the core of the system, and the first Mosaic software to appear was the XMosaic WWW client for UNIX with X. Other clients of similar functionality are under development for the Apple Macintosh and the PC with Windows.

The capabilities of the XMosaic browser include:

- o Support for NCSA's Data Management Facility (DMF) for scientific data.
- o Support for transferring data with other NCSA tools such asCollage, using NCSA's Data Transfer Mechanism (DTM).
- o The ability to "check out" documents for revision, and to check them back in again.
- o Local and remote annotation of Web documents.

Future planned functionality includes:

- o In-line non-text data (in addition to images).
- o Information space graphical representation and control.
- o Hypermedia document editing.
- o Information filtering.

NCSA intends to make the entire Mosaic system publicly available and distributable.

The XMosaic browser was used extensively for finding and retrieving information used to prepare this report.

Web Publishing

Making a web is as simple as writing a few SGML files which point to your existing data. Making it public involves running the FTP or HTTP daemon, and making at least one link into your web from another. In fact, any file available by anonymous FTP can be immediately linked into a web. The very small start-up effort is designed to allow small contributions.

At the other end of the scale, large information providers may provide an HTTP server with full text or keyword indexing. This may allow access to a large existing database without changing the way that database is managed. Such gateways have already been made into Digital's VMS/Help, Technical University of Graz's "Hyper-G", and Thinking Machine's WAIS systems.

There are a few editors which understand HTML - for instance on UNIX and on the NeXT platform.

Published non-text data

See the multimedia demo node on:

http://hoohoo.ncsa.uiuc.edu:80/mosaic-docs/multimedia.html

This contains links to images, sound, movies and postscript media types. The media type is determined by the filename extension in the URL specification of the target node. The (XMosaic) client uses this to invoke a separate program appropriate for displaying the media type, or in some cases it can be displayed embedded within the source document. The latter method uses an <IMG> tag, which is part of HTML2.

Advantages

WWW is a hypertext system and its underlying technology is thus richer than Gopher. The use of SGML, which is of increasing importance in hypermedia systems, allows a great deal of expressiveness and structure, and enables text to be presented in an attractive way. The facilities for multimedia data in the extended versions of HTTP and HTML are excellent. It also seems that QOS and management issues identified in Chapter 2 are to some degree catered for in these extensions.

Limitations

There is no indication in the source anchor of the media type of the destination node, or of its size (this has been ruled out on the argument that the information is likely to degrade with time). It is necessary to perform a HEAD request (in HTTP2) to deduce this.

Link source anchors must be in text documents, so non-text nodes must be leaf nodes. However, with HTML2 using the <IMG> tag, an embedded bitmap may be used as a source anchor, and the position of the mouse click within the image is passed to the server, which can then choose to return a different document depending on where in the image the mouse was clicked.

WWW is much less prevalent than Gopher, partly because of an (erroneous?) perception that setting up an HTTP server is more complex than setting up a Gopher server. There are only about 60 servers world-wide; however the growth in the use of WWW is much faster than the growth in the use of Gopher. The availability of sophisticated WWW clients such as XMosaic is fuelling this growth.

3.4. Evaluating Existing Tools

This section compares the capabilities of the Gopher, WAIS and WorldWide Web systems (abbreviated as GWW) to the informal requirements defined in section 2.3.

Platforms

The table below gives the names of the most important client software for each of GWW on the three most important platforms of interest. WWW is the weakest, with clients for the Macintosh and the PC still under development. The main PC Gopher client is "PC Gopher III", which is a DOS program, not a Windows program.

CLIENTS	Gopher	WAIS	WWW
Macintosh	TurboGopher	WAIStation	(No name) (beta version available)
PC with Windows	HGopher (two others also available)	WAIS for Windows, WAIS Manager	Cello (beta version available), Mosaic (beta due 3Q93)
UNIX with X	Xgopher, XMosaic	XWAIS	XMosaic

At present, multimedia support in most of these clients (where it exists) is limited to the invocation of external "viewer" programs for particular media types. The exception is XMosaic, which supports in-line images in WWW documents.

Media Types

The GWW tools can all handle multiple media types well.

- o Text is very well supported by all three tools. WWW offers facilities for displaying "richer" text, supporting headings, lists, emphasised text etc., in a standardised way.

- o Image data is also well supported, using either external viewers (e.g., the TurboGopher client software on a Macintosh might invoke the JPEGView program to display an image); or in-line display within a text document (WWW with XMosaic on UNIX).

- o There is little direct support for application-specific data, but most systems allow data of a nominated type to be passed to an external viewer or editor program. This tends to be a function of the client software rather than being built in to the protocol or server. There has been discussion in the WWW community about using TeX for representing mathematical equations, and about providing "panels" within a text document where a separate application could render its application-specific data (or indeed any data which can be represented spatially). This latter suggestion fits well with the OLE (Object Linking and Embedding) approach used in Microsoft Windows.

- o Sound can be supported through the external "viewer" concept. Some platforms don't have readily-available "viewers" with "tape recorder"-style controls for replaying. There is no single commonly-accepted sound encoding format.

- o Video data can be handled using external viewers. MPEG and QuickTime are the most common encodings.

One essential capability of a client/server protocol is the ability for the client to determine the type of a node (and a list of available encodings) before downloading it. WAIS and Gopher transfer this information in the result set and menu respectively. WWW clients currently determine this information either from analysing the URL of a target node, or by the occurrence of the <IMG> tag. The new WWW HTTP2 protocol allows the media type and encoding of a node to be determined through a separate interaction with the server.

The GWW systems all use different methods for expressing type and encoding. WAIS does not distinguish the encoding from the media type. WWW is moving to the MIME type/encoding system. Gopher does not distinguish type and encoding, but Gopher+ does, and is also moving to the MIME type/encoding system.

Hyperlinks

Only the WWW system has hyperlinks. Source anchors may be text, images, or points within an image. Target anchors may be entire nodes of any media type, or points within (with HTTP2, portions of) text nodes.

Gopher+ could potentially be enhanced to include hyperlinks, but there seems to be no development effort going towards this - those who need hyperlinking are using WWW.

Gopher menus can be constructed to allow alternative views of gopherspace. For instance, a geographically-organised menu tree of gopherspace is in place, but a parallel subject-based menu tree could be added as an alternative way of access to the same data. (There are in fact moves to set this up.) Since WWW offers a superset of Gopher functionality, these comments also apply to the Web. In fact, the Web already has a rudimentary subject tree.

In both Gopher and WWW, non-textual data may be used in different information structures without having to maintain more than one copy.

Presentation

There is little support in GWW for controlling the presentation of non-text data.

- o Backdrops are not supported by GWW.

- o Buttons are supported in a limited way - typically, a node is retrieved by clicking on a highlighted text phrase, or on an entry in a list. In XMosaic, bitmap images can be used as buttons. However, there is no support for different styles of button. Client software may have generic navigation buttons (e.g., "Back", "Next", "Home") which are always available and don't form part of a node.

- o Synchronisation in space is not supported by GWW, except that WWW supports contextual synchronisation of images using the <IMG> tag.

- o Synchronisation in time is not supported by GWW.

Searching

WAIS supports keyword searching, and is very well suited for that task. The Gopher+ protocol could potentially support multimedia database querying applications through the ASK attribute, but there is as yet no server implementation which supports such database applications. In the WWW project, there are ongoing discussions on how best to extend HTML to cope with database query applications - an <INPUT> tag has been suggested - but no consensus has yet emerged.

Both Gopher and WWW can make use of WAIS-type keyword searching: either by incorporating WAIS code into the server (enabling WAIS index files to be searched); or through WAIS gateways, which run searches on remote WAIS servers in response to queries from non-WAIS clients.

Interaction

XMosaic allows users to make text (or on some platforms, audio) annotations to any text node. The annotations appear at the end of the text display.. They are held locally - other users of the node do not see the annotations (but a recently added facility allows globally-visible annotations held on an "annotation server"). Text annotations may include hyperlinks to other nodes (provided the user knows how to use HTML). Other clients do not provide such facilities.

There is a move to add an "email" address notation to URL. This would allow WWW client software to invoke a mail program when a user selects an anchor with such a URL.

There are plans to allow WWW users to delineate a rectangular area of interest within an image for use in an HTTP request.

There is no support in GWW clients for interacting with sequences of images in the way described in section 2.3.6.

Quality of Service

The user expectations for responsiveness mentioned in section 2.3.7 are difficult to meet with currently-deployed wide-area network (or even LAN) technology, particularly for voluminous multimedia data. None of the GWW systems currently exploit the emerging isochronous data transfer capabilities of protocols such as RTP and technologies such as ATM. None of them make serious attempts to alleviate the problem in other ways (except for WWW, which defines some mechanisms in HTTP2 for format negotiation based on size and available bandwidth considerations).

Management

The following table shows the support for three key management facilities in the GWW systems. The first two facilities require support in the client/server protocol, the third requires support in the server, but depends on authentication being available.

	Gopher	WAIS	WWW
Access control and authentication	No	No1	Yes, in HTTP2

Charging support	No	No	Yes, in HTTP2
Monitoring for statistical and assessment purposes	No	No	No

Note:

1. "Access-control-facility" is a feature of Z39.50 which is not used by the current WAIS implementations.

Scripting Requirements

None of the GWW systems have facilities for the execution of scripts by the client, because of security issues (it would be too easy for a malicious "trojan" script to be executed). Gopher and WWW servers have the ability for a UNIX script to be run by the server, with the script output returned to the client. Scripting as understood in the context of stand-alone multimedia applications does not exist in GWW.

Bytestream Format

None of the three GWW systems use a bytestream format for interchanging collections of material. There has been some talk about setting up a system akin to the "Trickle" mail server, for retrieving single document nodes from GWW using mail. Such a system has been implemented for WWW.

Authoring tools

Gopher is sufficiently simple to set up that no special authoring tools are required. WAIS requires only an indexing program (as discussed in section 3.2) for preparing material for publication.

WWW, because it uses a sophisticated authoring language (HTML), benefits from the availability of authoring tools. There are HTML editors for UNIX (using the tk toolkit) and the NeXT system. There are no authoring tools designed specifically for exploiting the multimedia capabilities of WWW, mainly because these capabilities are still evolving.

4. Research

This section describes some current research projects in the area of distributed hypermedia information systems.

4.1. Hyper-G

Hyper-G [10] is an ambitious distributed hypermedia research project at a number of institutes of the IIG (Institutes for Information-Processing Graz), the Computing and Information Services Centre of the Graz University of Technology, and the Austrian Computer Society. It is funded by the Austrian Ministry of Science. It combines concepts of hypermedia, information retrieval systems and documentation systems with aspects of communication and collaboration, and computer-supported teaching and learning.

Unlike WWW, Hyper-G supports bi-directional links. This enables users to see which other documents reference the one they are using, and also allows the system to avoid dangling pointers when a linkedto document is deleted. Another difference from WWW is that links are kept separately from their source and target nodes, to allow easy linking of read-only documents and for ease of link maintenance. In addition to manually defined links, Hyper-G supports automatic static and dynamic (i.e., view-time) generation and maintenance of links.

Hyper-G has a concept of generic "structures" - an additional layer of relationships imposed on (and orthogonal to) the web of documents and links. A document can be part of more than one structure, and structures may be hierarchically related. Types of structure include:

- o "Clusters" are a set of documents which are all presentedtogether.
- o "Collections" are unordered sets of documents or other structures, and can be used as query domains or to construct gopher-like menus.
- o "Paths" are ordered sets of documents or structures, which must be visited sequentially.

One application of the structure concept is the provision of "guided tours" through the information space.

In addition to hypernavigation, the collection hierarchy and guided tours, another strategy for interaction with the system is the use of database queries. Two kinds of query are supported: keyword searching in a user-defined list of databases; and collection specific form-filling queries. In the latter case, the answer to the query may appear dynamically as the form is filled out.

Four modes of user identification are supported: "identified", where a userid is publicly associated through name and address information with a particular individual; "semi-identified", where a userid is associated by the system with an individual, but the user is only known to other users through a pseudonym; "anonymously identified", where the userid is not associated by the system with any individual; and "anonymous", where there is no userid (or a generic userid such as "guest"). Possible operations in the system depend on the user's mode of identification. Users may access the system in any desired mode, and switch to other modes only when necessary.

Hyper-G contains specific support for multilingual documents and document clusters. Users may specify an ordered list of preferred languages, for instance. There are plans to experiment with automatic translation programs.

Integration of other, external, systems such as WWW into Hyper-G in a seamless manner is possible.

Hyper-G is in use as a CWIS within Graz Technical University. Client software is available for UNIX workstations from DEC, HP, SGI, and SUN. The system is still in an experimental state, but it has been used by about 200 students as part of a course on the social impact of information technology.

4.2. Microcosm

Microcosm [11] is an open hypermedia system developed at the University of Southampton. It is implemented on the PC under MS Windows, and versions for the Apple Macintosh and for UNIX with X are under development.

Microcosm consists of a number of autonomous processes which communicate with each other by a message-passing system. Information about hyperlinks between documents is stored in a link database, or "linkbase", and is not stored in the documents themselves. This has the advantages that:

- o Links to and from read-only documents (perhaps stored on CD-ROM) are possible.
- o Documents need undergo no conversion process to be imported into the system - they can still be viewed and edited using the original application which created them, without the link information getting in the way.

- o It is as easy to establish links to and from non-text documents as text documents.

In Microcosm, the user interacts with a "viewer" program for a particular media type. Such programs may be specifically written for use with Microcosm (about 10 such viewers have been written for a number of common media types and encodings); or they may be a program adapted for use with Microcosm (the programmability of Microsoft Word for Windows has allowed it to be so adapted); or it may even be a program with no knowledge of Microcosm.

The user selects an object (e.g., a piece of text) in the viewer, and requests Microcosm to perform an action with the object - typically to follow a link to another document. This may involve executing another viewer to display the target document.

Microcosm link source anchors may be specific (denoting a unique point in a particular document), local (denoting any occurrence of a particular object in a particular document) or generic (denoting any occurrence of an object in any document). Target anchors may specify specific objects within a document. Other link styles are textretrieval links (looking up a full-text index , as WAIS does), and relevance links to a set of documents using similar vocabulary to the source document (again, similar to WAIS's relevance feedback).

Links may be created by readers as well as by authors. Dynamically computed links may be added to the permanent linkbase for later use. A history of link traversal is maintained, and "guided tours" may be established through the system which allow the reader to stray from and return to the tour.

Microcosm viewers operate by sending messages to the Microcosm system. In MS Windows, these messages are transferred using DDE (Dynamic Data Exchange); in the Apple Macintosh version Apple Events are used, and sockets are used on UNIX. For viewers which are not Microcosm aware, the user must transfer the selected object to the system clipboard before being able to follow a link from it.

Networking support in Microcosm is currently under development. Components of Microcosm may be distributed to multiple machines there is not necessarily a concept of "client" and "server".

There are problems with the Microcosm approach, common to systems which maintain link information separately from documents, and which use external viewers.

- o Documents move and change, thus invalidating links. Microcosm datestamps links to help to detect (but not correct) such problems.

- o It is not always clear what links are available to be followed from a document, since the viewer program is unaware of the contents of the linkbase.

- o It is not always possible to indicate the object within a document which is the target anchor of a link. Many viewers automatically show the start of the document (e.g., a word processor), or perhaps the entire document (e.g., a picture viewer). The user has no way of knowing which part of the target document the link just followed points to.

Microcosm may be viewed as an integrating hypermedia framework - a layer on top of a range of existing applications which enables relationships between different documents to be established.

Microcosm is currently being "commercialised".

4.3. AthenaMuse 2

AthenaMuse 2 (AM2) is an ambitious distributed hypermedia authoring and presentation system under development by the AthenaMuse Software Consortium based at MIT. It is based on the earlier AM1 system developed as part of MIT's Project Athena. The first version of AM2 is scheduled for January 1994, and will be "pre-commercial software", with a fully-commercialised version due about 6 months later. Both the educational and commercial sectors are the intended market. The system will initially be based on X and UNIX workstations, but PC/Windows will also be supported in a second phase. Apple Macintosh support has a lower priority.

The specifications of AM2 are available in [12]. Some of the key points are:

- o AM2 will support import and export of application from and tostandard forms. The project is watching standards such as HyTime, MHEG and ODA.

- o Several "application themes", or frequently-occurring collections of functionality, are viewed as useful. These are as follows:

Application Theme	Interactive?
Presentation of multimedia data	No
Exploration of a rich multimedia	Yes

environment	
Simulation of a real-world scenario	Partially
Communication of real-time information to the user	No
Authoring	Yes
Annotation of material	Yes

- "Interface templates" allow a multimedia application to make use of a common format for presenting a range of content. This is similar to the "backdrop" concept mentioned in section 2.3.4.

- A range of link types will be supported.

- Media content editors and interface/application editors for structuring will be provided. A third class of editor, the "hypermedia notebook", will allow readers to excerpt and annotate media from AM2 applications.

The project is developing multimedia network services, including the transmission of digital video, using a client-server paradigm.

4.4. CEC Research Programmes

Some of the research programmes sponsored by the Commission for the European Community (CEC) contain apparently relevant projects. [1] has further details of some of these projects.

RACE programme

The RACE programme is outlined in [13], which should be consulted for further information about the projects described below. The RACE programme targets the industrial, commercial and domestic sectors, and results are not necessarily directly applicable to the research and academic community. RACE project numbers are given.

RACE Phase I projects, which have mostly completed:

R1038 MCPR - Multimedia Communication, Processing and Representation. This project developed a demonstrator multimedia system with communications capability for travel agents.

R1061 DIMPE - Distributed Integrated Multimedia Publishing Environment. The project designed and implemented interim services for compound document handling, and defined a distributed publishing architecture.

R1078 European Museums Network. This project aimed to demonstrate interactive navigation through a pool of multimedia museum objects, using ISDN as the communications network.

RACE Phase II projects:

R2008 EuroBridge.

Aims to demonstrate multi-point multimedia applications running over DQDB, FDDI and ATM test networks.

R2043 RAMA - Remote Access to Museum Archives

This project follows on from R1078.

R2060 CIO - Coordination, Implementation and Operation of Multimedia Services.

One aspect of this project is JVTOS - a "Joint Viewing and Teleoperation Service". This aims to integrate standard multimedia applications running on a range of heterogeneous machines into a cooperative working environment, allowing individuals to view and interact with multimedia data on colleague's machines.

ESPRIT Programme

The ESPRIT research programme is outlined in [14], which should be consulted for further information about the projects listed below. ESPRIT project numbers are given.

28 MULTOS - A Multimedia Filing System

This project, which ran from 1985 to 1990, developed a client/server system for filing and retrieval of multimedia documents using the ODA interchange format standard (ODIF).

5252 HYTEA - HyperText Authoring

This project, which runs from 1991 to 1994, aims to develop a set of authoring tools for large and complex hypermedia applications.

5398 SHAPE - Second Generation Hypermedia Application Project

This project is developing a portable software environment comparable to a CASE tool intended to facilitate the realisation of complex hypermedia applications.

5633 HYTECH - Hypertextual and Hypermedial Technical Documentation This project, which ran from 1990-1991, was to assess the feasibility of hypermedia technology and to devise needed extensions to it in order to support applications dealing with technical documentation management.

6586 PEGASUS - Distributed Multimedia Operating System for the 1990s This project is aimed at the design of an operating system architecture for scalable distributed multimedia systems and the development of a validating prototype, the design and implementation of a distributed complex-object service and a global name service, the development of mechanisms for the creation, communication and rendering of fully digital multimedia documents in real time and in a distributed fashion, and the design and implementation of an application for the system: a digital TV director.

6606 IDOMENEUS - Information and Data on Open Media for Networks of Users. This project, which started January 1993, brings together workers in the database, information retrieval, networking and hypermedia research communities in the development of an "ultimate information machine". It "will coordinate and improve European efforts in the development of next-generation information environments capable of maintaining and communicating a largely extended class of information on an open set of media". Because of the close match between the subject of the IDOMENEUS project and the RARE WG-IMM, it is recommended that RARE establish a liaison with this project.

4.5. Other

Some other research projects of less immediate relevance are listed below. Some of these projects are described further in [1].

- o Xanadu is a project to develop an "open, social hypermedia" distributed database server, incorporating CSCW features. It has been in existance for many years and has been funded by a number of companies. The current status of this project is not known, and although iminent availability of alpha-test versions has been announced more than once, no software has been delivered.

- o CMIFed [15] is an editing and presentation environment for portable hypermedia documents being developed at CWI, Amsterdam, NL. It is based on the "Amsterdam Model" of hypermedia [16], which is an extension of the Dexter hypertext reference model incorporating "channels" for media delivery and synchronisation constraints.

- Deja Vu [17] is a proposed "intelligent" distributed hypermedia application framework. It is intended as a vehicle for research in the areas of: hypermedia systems, object-oriented programming, distributed logic programming, and intelligent information systems. Proposed techniques for use in the Deja Vu framework include "inferential links", defined automatically according to predefined rules. A scripting language for use both by information providers and users is planned. This project is at a very early (proposal) stage, and as yet relatively little software has been developed. Deja Vu is intended principally as a research framework rather than as a service tool.

- Demon is a project at Bellcore, US, investigating the network requirements of near-term residential multimedia services. The project is designing and implementing an experimental application which serves the needs of casual multimedia users.

- InfoNote is a distributed, multiuser hypermedia system from Japan, implemented on a NEC EWS4800 running UNIX and X. InfoNote has an editor which can create Japanese texts, figures, and raster images. The same windows are used both for editors and browsers. The functionality of the window can be changed at any time if data is not write-protected.

- MADE - Multimedia Application Demonstration Environment - is a project at British Telecom's research laboratory which centres on the use of the developing MHEG standard to access a multimedia object server. The server platform is a Sun SPARCstation with an object-oriented database package (ONTOS). Audio, video, text and graphical media types are covered. The University of Kent is working on a sub-project: "Multi-user Indexing in a Distributed Multimedia Database".

- Zenith aimed to establish a set of principles to assist designers and developers of object management systems intended for distributed multimedia design environments. The project implemented a prototype generalised multimedia object management system.

5. Standards

5.1. Structuring Standards

This section describes some of the important standards for providing hyperstructure to multimedia data.

SGML

SGML (Standard Generalized Markup Language - ISO 8879) is a metalanguage for defining markup notations for text. SGML is used to write Document Type Definitions or DTDs, to which individual document instances must conform. It finds application in a wide and increasing range of text processing applications.

The relevance of SGML to distributed hypermedia systems is surprisingly high, mainly because of the great expressive power of SGML, and its ability to handle non-textual data using "external entities" and "notations".

- The World-Wide Web is an SGML application with its own DTD.

- The important HyTime hypermedia structuring standard (see below) is based on SGML.

- The forthcoming MHEG hypermedia structuring standard (see below) has an SGML encoding.

- SGML has been used in research hypermedia systems - for example Microcosm.

- SGML is used in some commercial hypermedia systems - for example DynaText.

- SGML is of increasing importance for academic publishing houses.

It was interesting to note that at a recent (CEC-sponsored) workshop on Hypertext and Hypermedia standards, most of the speakers were conversant with and supportive of the use of SGML for such systems.

A related standard which may become important for SGML on networks is SDIF (SGML Data Interchange Format - ISO 9069). This standard specifies how an SGML document, which may exist in a number of separate files of different media types, may be encoded using ASN.1 into a single bytestream. The entity structure is preserved, so that the bytestream may be decoded by the recipient into the same set of files.

HyTime

HyTime (Hypermedia/Time-Based Structuring Language) is a standardised infrastructure for the representation of integrated, open hypermedia documents. It was developed principally by ANSI committee X3V1.8M, and was subsequently adopted by ISO and published as ISO 10744.

HyTime is based on SGML. It is not itself an SGML DTD, but provides constructs and guidelines ("architectural forms") for making DTDs for describing Hypermedia documents. For instance, the Standard Music Description Language (SMDL: ISO/IEC Committee Draft 10743) defines a (meta-)DTD which is an application of HyTime. In fact, HyTime started as an attempt to produce a markup scheme for music publishing purposes.

HyTime specifies how certain concepts common to all hypermedia documents can be represented using SGML. These concepts include:

- association of objects within documents with hyperlinks
- placement and interrelation of objects in space and time
- logical structure of the document
- inclusion of non-textual data in the document

An "object" in HyTime is part of a document, and is unrestricted in form - it may be video, audio, text, a program, graphics, etc. The terminology used in HyTime (and in this section) thus differs slightly from the terminology used in the rest of this report. A HyTime object corresponds roughly to a node as defined in section 1.2, and a HyTime document is a hyperdocument in the terminology of this report.

HyTime consists of six modules, which are very briefly and selectively described below:

- Base module. This provides facilities required by other modules, including a lexical model for describing element contents; facilities for identifying policies for coping with changes to a document, or traversing a link ("activity tracking"); and the ability to define "container entities" which can hold multiple data objects. This last was added to the HyTime standard at a late stage, at the instigation of Apple Computers Inc, as a "hook" for their Bento specification [18].

- Measurement module. This allows for an object to be located in time and/or space (which HyTime treats equivalently), or any other domain which can be represented by a finite coordinate space, within a bounding box called an "event", defined by a set of coordinate points. Coordinates may be expressed in any units (predefined units include femtoseconds, fortnights, millenia, angstroms, Northern feet and lightyears!).

- Location Address module. In addition to the fundamental ability of SGML to identify and refer to elements, this module provides a special "named location address" architectural form which can be used to refer indirectly to data which spans elements, or which is located in external entities. Data may also be addressed indirectly through the use of "queries", which return addresses of objects within some domain which have properties matching the query. A "HyQ" notation is provided for defining the query.

- Hyperlinks module. Two basic types of hyperlink are defined: the contextual link (clink) has two anchors, one of which is embedded in a document to explicitly denote the anchor location; and the independent link (ilink) which may have more than two anchors, and which does not require the anchors to be embedded in the document. ilinks thus allow hyperlink information to be maintained separately from document content.

- Scheduling module. This specifies how events in a source finite coordinate space (FCS) are to be mapped onto a target FCS. For instance, events on a time axis could be projected onto a spatial axis for graphical display purposes, or a "virtual" time axis as used in music could be projected onto a physical time axis.

- Rendition module. This allows for individual objects to be modified before rendition, in an object-specific way. One example is modification of colours in image so that it can be displayed using the currently-selected colour map on a graphics terminal, or changing the volume of an audio channel according to a user's requirements.

It is not envisaged that a hypermedia application would need to use the entire range of HyTime facilities. An application designer is able to choose appropriate HyTime architectural forms, and to add application-specific constraints to them. The designer may also of course use non-HyTime SGML elements and attributes, but these aspects of the application can't be understood by a "HyTime engine". Even in the absence of a HyTime engine, the HyTime architectural forms provide a useful base of ideas from which a hypermedia system designer may wish to work.

The role of a HyTime engine is not specified in the standard, but essentially it is a (sub)program which recognises HyTime constructs in document instances and performs application-independent processing on them. For instance, it could interact with multimedia network servers to resolve and access hyperlink anchors. A commercial HyTime engine (HyMinder) is under development by TechnoTeacher in the US, and the Interactive Multimedia Group at the University of Massachusetts - Lowell (contact lrutledg@cs.ulowell.edu) is also working on a HyTime engine (HyOctane).

The Davenport group (a loose consortium of interested companies and individuals) is producing a series of standards on hypermedia which further constrain the HyTime architectural forms. One example is the SOFABED module [19], which standardises the representation of certain kinds of navigational information - tables of contents, indexes and glossaries.

HyTime was envisaged as an interchange format rather than as a format for directly-executable hypermedia applications. It is therefore very expressive, but may be difficult to optimise for run-time efficiency.

An attempt has been made [20] to adapt the hyperlink structure in WWW's existing HTML DTD to comply with HyTime's clink architectural form. This requires changes to WWW document instances as well as to browser software, and in the absence of any immediate benefit it has found little favour with the WWW community. However, it is possible that HTML2 will use some aspects of HyTime.

It is recommended that any further RARE work on networked hypermedia should take account of the importance of SGML and HyTime.

MHEG

MHEG stands for the Multimedia and Hypermedia information coding Experts Group, also known as ISO/IEC JTC1/SC29/WG12 (it used to come under SC2). This group is developing a standard "Coded Representation of Multimedia and Hypermedia Information Objects" (ISO CD 13522, or CCITT T.171), commonly called MHEG. The standard is to be published in two parts - part 1 being the base notation, representing objects using ASN.1, and part 2 being an alternate notation which uses SGML. Part 1 has nearly (June 1993) achieved CD status, and is intended to reach full IS in 1994. Part 2 is intended to reach the CD stage in late 1993.

MHEG is suited to interactive hypermedia applications such as on-line textbooks and encyclopaedia. It is also suited for many of the interactive multimedia applications currently available (in platformspecific form) on CD-ROM. MHEG could for instance be used as the data structuring standard for a future home entertainment interactive multimedia appliance. Telecommunications operators are interested in MHEG for providing interactive multimedia services across ISDN.

To address such markets, MHEG represents objects in a non-revisable form, and is therefore unsuitable as an input format for hypermedia authoring applications: its place is perhaps more as an output format for such tools. MHEG is thus not a multimedia document processing format - instead it provides rules for the structure of multimedia objects which permits the objects to be represented in a convenient "final" form with the aim of direct presentation.

The MHEG draft standard is expressed in object-oriented terms. The main object classes are outlined briefly below.

- o Content class. A content object contains the encoded (monomedia) information to be presented, along with attributes which identify the type of information and the encoding method, and mediaspecific attributes such as fonts used, sampling rate, image size, etc.

- o Selection class and Modification class. The user may interact with MHEG objects which inherit interactive behaviour from these classes. (The MHEG object model supports multiple inheritance.)

- o Action class. Two types of action may be applied to objects: projection, which controls how objects are rendered; and status actions which affect the state of objects.

- o Link class. MHEG hyperlinks connect a "start" object with one or more "end" objects. Links consist of a set of conditions relating to the state of the start object, and a set of actions which are carried out when these conditions are satisfied. Links also define the spatio-temporal relationships between objects.

- o Script class. Script objects are used to describe more complex interobject linkages (e.g., multiple-source links). MHEG does not define a scripting language - instead it provides a formalism for encapsulating scripts which may be executed by an external program (see SMSL below).

- o Composite class. Related objects may be grouped together into a single composite object (recursively). The relationships between content objects within a composite object are determined by link and script objects which also are members of the composite object.

- o Descriptor class. Descriptor objects contain general information about sets of interchanged objects, so that a target system can ensure it has adequate resources to run the hypermedia application represented by the object set.

The relationship between HyTime and MHEG has not yet been fully established. One possible relationship [21] is that an MHEG application could be the output of a compilation process which used an equivalent HyTime document as input. This approach would benefit both from the expressive power of HyTime and the run-time efficiency of MHEG. However, it has yet to be shown that this is feasible, since the capabilities of HyTime and MHEG do not completely overlap.

There seems to be relatively little interest in or awareness of MHEG within the Internet community, which is only just beginning to be aware of HyTime. In view of the draft nature of the MHEG standard, this report recommends that RARE should not invest substantial effort in MHEG at this time. However, particularly in view of the interest in it shown by PTTs, a watching brief should be kept on MHEG, as it may well be relevant in the future.

ODA

The Open Document Architecture standard (ODA - ISO 8613 or T.140) is a compound document interchange format designed for transferring documents between open systems. It is able to represent documents in both a formatted form and a processable (i.e., revisable) form, thus allowing both the content and the printed appearance of the document to be unambiguously transferred.

In addition to text data, ODA supports graphics and image data. A revised version to be published in 1993 will support colour. Future developments include support for audio content (underway) and video content (planned). An interface to MHEG is also planned.

ODA differs from SGML in that the former concerns itself with the physical appearance of the document, while SGML deliberately avoids doing so. SGML concerns itself with semantic markup, and can be used to describe a wide range of data and document architectures. ODA has a more limited concept of a document.

Hypermedia extensions to ODA (HyperODA) are underway. The extensions will support:

- o References to data held externally to the document (similar to SGML's external entities?).

- o Non-linear structures, using contextual and independent hyperlinks based on the HyTime model.

- o Temporal relationships between document components (e.g., sequential, parallel, cyclic, duration, start delay).

HyperODA is not being developed in competition to HyTime or MHEG its purpose is to add hypermedia features to ODA rather than to be a completely general framework for hypermedia applications.

Bearing in mind that:

- o the HyperODA extensions are still under development;

- o in some senses ODA can be seen as a competitor to SGML, which has greater presence in the hypermedia world;

- o there seems to be a lack of enthusiasm for ODA in the Internet community (the IETF WG on piloting ODA has disbanded);

- o Adobe's newly-released Acrobat technology (described below) will have a significant effect on the marketplace;

this report recommends that ODA should not form a basis for investment in networked hypermedia technology by RARE.

PREMO

PREMO (Presentation Environment for Multimedia Objects) is a new work item in ISO/IEC JTC1/SC24 (the graphics standards subcommittee). An initial draft [22] exists, and the schedule calls for a CD by June 1994, a DIS by June 1995, and the final IS by June 1996.

PREMO addresses the construction of, presentation of, and interaction with multimedia objects. It specifies techniques for creating audiovisual interactive single and multiple media applications. It is consistent with the principles of the Computer Graphics Reference Model (CGRM, ISO 11072), and is defined in object-oriented terms.

It is not clear how PREMO relates to HyTime and MHEG. Although these standards are listed in section 2 (References) of the initial draft, they appear not to be mentioned in the text. The wisdom of developing what appears to be yet another structuring standard for multimedia data is doubtful.

The PREMO work is not sufficiently advanced to permit a judgement of its usefulness in satisfying the requirements under discussion.

Acrobat

Adobe, Inc. has introduced a new format called Acrobat PDF, which it is putting forward as a potential de facto standard for portable document representation. Based on the Postscript page description language, Acrobat PDF is also designed to represent the printed appearance of a document (which may include graphics and images as well as text. Unlike postscript however, Acrobat PDF allows data to be extracted from the document. It is thus a revisable format. It includes support for annotations, hypertext links, bookmarks and structured documents in markup languages such as SGML. PDF files can represent both the logical and the formatting structure of the document.

Acrobat PFD thus appears to offer very similar functionality to ODA. Adobe's successful Postscript de facto standard profoundly influenced information technology - it is possible that if successful, Acrobat PDF will be almost as important. RARE should be aware of this technology and its potential impact on multimedia information systems.

5.2. Access Mechanisms

This section describes some standards which are useful in providing network access to multimedia data. Of course, there are many multimedia transport protocols, which this report does not attempt to describe (see [1] for further information). The protocols mentioned below are search/retrieve protocols which were not mentioned in [1].

Multimedia Extensions to SQL

A new work item in ISO (ISO/IEC JTC1 N2265) to extend the SQL standard to include multimedia data is expected to be approved shortly. Initially this work will concentrate on developing a framework, and on free text data. Support for non-text data will be added later, using a separate part of the standard for each media type.

The expected timescale for this standardisation work is lengthy (part 1 - the framework - is targeted for completion in 1996).

There are suggestions that this standard could be used as a query language in conjunction with the HyQ query component of the HyTime standard.

DFR

DFR is the Document Filing and Retrieval system, specified in ISO 10166-1 and ISO 10166-2. It is intended for office automation applications, and falls within the Distributed Office Applications (DOA) model of ISO 10031-1. DFR has design similarities to the ISO Directory and to the X.400 Message Store, and it is likewise part of OSI.

DFR defines a Document Store, which provides a service to a DFR User over an OSI protocol stack incorporating ROSE (and optionally RTSE). A document in the Document Store may have a number of attributes associated with it, including pointers to related documents. There is support for multiple versions of the same document, and for hierarchical groups of documents. The access protocol supports searching for documents based on their attributes. DFR itself does not restrict the content of documents in any way, but the natural partner to DFR is the ODA standard for document content.

It is not clear that DFR offers significantly more useful functionality than is available from other, simpler access protocols already in use on the Internet.

5.3. Other Standards

This section briefly describes other standards in this area and discusses their relevance.

MIME

MIME (Multipurpose Internet Mail Extensions) is a mechanism for transferring multimedia information in an RFC822 mail message. STD 11, RFC 822 defines a message representation protocol which specifies considerable detail about message headers, but which leaves the message content as flat ASCII text. RFC 1341 redefines the format of message bodies to allow multi-part textual and non-textual message bodies to be represented and exchanged without loss of information. Because RFC 822 said very little about message content, RFC 1341 is largely orthogonal to (rather than a revision of) RFC 822.

MIME provides facilities to include multiple objects in a single message, to represent text in character sets other than US-ASCII, to represent formatted multi-font text messages, to represent non textual material such as images and audio fragments, and generally to facilitate later extensions defining new types of Internet mail for use by co-operating mail agents. It does not define any structure to allow relationships between body parts within a message to be expressed.

For the purposes of the requirements considered by this report, the relevance of MIME is that it separates media type from media encoding, and that it defines a procedure for registering values of these attributes.

The MIME construct of chief interest is the "Content-Type" field. This contains a MIME "type" and "subtype", and any "parameters" which further qualify the subtype. The register of MIME content-types is maintained by the Internet Assigned Numbers Authority (IANA). Content types defined in the MIME standard itself include:

Type	Subtype	Parameters	Meaning
text	plain	charset	Plain text
	richtext	charset	Text with SGML-like markup for representing formatting.
image	jpeg		JPEG File Interchange Format
	gif		Graphics Interchange Format
audio	basic		8-bit -law 8kHz PCM encoding
video	mpeg		
application (used for application -specific data)	ODA	profile (Document Application Profile)	Open Document Architecture document.
	octet-stream	name (e.g., filename); type (for human recipient), etc.	General binary data such as an arbitrary binary file.
	postscript		Document in postscript.

Private experimental values of types and subtypes starting with X may be used between consenting adults without registration with IANA.

MIME also defines a "Content-Transfer-Encoding" field, which is used to specify an invertible mapping between the "native" encoding of a media type and a representation that may be readily exchanged using 7bit mail transfer protocols.

WWW's HTTP2 protocol makes use of MIME media type and encoding attributes, and also uses MIME's message format for retrieving data from the server. It is the first MIME application to utilise the 8bit Content-Transfer-Encoding, which essentially means no encoding.

SMSL

SMSL is the Standard Multimedia Scripting Language. It is a proposed new work item for ISO/IEC JTC1/SC18/WG8 (HyTime) and JTC1/SC29/WG12 (MHEG). The functional requirements are expected to be completed in 1994, and the coding scheme completed in 1995.

SMSL is designed as an open language with a similar purpose to existing vendor-specific scripting languages such as Macromind's "Lingo", Kaleida's "Script/X", and Gain's "GEL". The intention is to offer an intermediate open multimedia scripting language which could be used both for interchange purposes, and for controlling the presentation of HyTime or MHEG multimedia structures. Several different approaches to defining SMSL have been suggested, including using the ANDF (Architecture-Neutral Distribution Format) approach, and basing SMSL on SGML or on the Scheme language.

The SMSL work is not sufficiently advanced to permit a judgement of its usefulness in satisfying the requirements under discussion. However, it is interesting to note that despite the descriptive power of HyTime and MHEG, there is still perceived to be a role for procedural scripting.

AVIs

The CCITT is defining a set of Audio Visual Interactive Services (AVIs), intended for offering to domestic and business consumers over a national network (e.g., by PTTs). These services will be specified as T.17x recommendations, and will include MHEG. These services would also make use of the SMSL work.

Insufficient information is available about this area to allow its relevance to be judged.

5.4. Trade Associations

This section mentions some trade associations which are involved in standards making in the multimedia area.

Interactive Multimedia Association

The Interactive Multimedia Association (IMA) is an international trade association with over 250 members, representing a wide spectrum of multimedia industry players. Members include Apple, Microsoft, MIT CECI (the developers of AthenaMuse 2), 3DO, and many other important market actors.

In 1989, the IMA initiated a "Compatibility Project", tasked with developing technical solutions to the cross-platform compatibility problem. The Project has published two important documents:

- o "Recommended Practices for Multimedia Portability" [23] outlines a specification for a common interface to be used by interactive video delivery systems. It has been adopted by the US Military as part of Military Standard 1379.

- o "Recommended Practices for Enhancing Digital Audio Compatibility in Multimedia Systems" [24] defines four standard digital audio data types and four sampling rates (from low-end -law 8kHz mono encoding, up through ADPCM modes to CD-quality 44kHz 16-bit stereo).

Work is continuing to produce further recommendations on other issues.

The Compatibility Project has now initiated a procurement process by publishing three Request for Technology (RFT) documents, defining the requirements of a platform-independent interactive multimedia system, including networking requirements. The RFTs cover "Multimedia System Services", a "Scripting Language for Interactive Multimedia Titles", and "Multimedia Data Exchange". An "Architecture Reference Model" for cross-platform desktop and distributed multimedia systems provides the framework for these RFTs, which are pragmatic documents outlining the technical requirements for time-based media handling in detail. Note that relatively little is said about non-time-based data.

A first reading of the Multimedia Data Exchange RFT reveals that the Apple Bento standard [18] and the Microsoft/IBM RIFF format [25] both influenced the development of this document. The selected system may well be based on one or both of these technologies.

A joint response to the Multimedia System Services RFT has been received from HP, IBM and Sun. Two responses to the Scripting Languages RFT have been received - from Kaleida (Script-X) and Gain Technology (GEL). Two partial responses to the Multimedia Data Exchange RFT have been received from Apple (Bento) and Avid (Open Media Framework).

Responses to the RFTs are currently being analysed by the IMA, and the result will be announced in November 1993. The specifications which will eventually result from this process will be important for future commercial multimedia products. It is important that the community keep a watching brief on the IMA Compatibility Project and its possible implications for distributed multimedia applications on the Internet.

Multimedia Communications Forum

The Multi-Media [sic] Communications Forum (MMCF) is a recently formed (June 1993) trade consortium whose initial members include IBM, National Semiconductor, Apple, Siemens and AT&T. Intended to complement the work of the IMA, the MMCF plans to develop guidelines and recommendations for the industry to help ensure "end-to-end network interconnectivity of multimedia applications, workstations and devices". They also plan to provide input to standards bodies.

It is still too early to say whether this forum will succeed. If the IMA Compatibility Project specifications, when they are published, leave networking issues open, then MMCF could have an important role to play. It is recommended that RARE consider becoming an Observing

Member ($350 US pa), entitling it to attend general and annual MMCF meetings (but not committee meetings), and to receive minutes and other general papers (but not working documents); with the prospect of becoming an Auditing Member ($1200 US pa) later if relevant.

Multimedia Communications Community of Interest

This is a very new organisation formed at a meeting in France in June 1993. Its charter is to promote the use of applications which let people in different locations view documents, images, graphics and full-motion video on a PC screen. The remit includes CSCW aspects. Members of the organisation include IBM, Intel, Northern Telecom, Telstra (Australia), BT, France Telecom and DB Telekom. The companies plan field trials of multimedia services in 1Q94.

6. Future Directions

6.1. General Comments on the State-of-the-Art

Distributed hypermedia systems are now emerging from the research phase into the experimental deployment stage. Every project team (and standards committee), almost without exception, hopes for their system to become the de facto standard for hypermedia.

As we've seen, Gopher and WWW already offer multimedia capability, but they are still largely oriented to the use of external viewers for non-text nodes. This "unintegrated" approach is in contrast to typical stand-alone multimedia applications, where the presentation of related information in different media is tightly integrated. The in-line image feature of XMosaic and the new version of HTML currently under development may represent the start of a move towards greater integration of different media in such distributed hypermedia systems.

Three important factors in the design of distributed hypermedia systems appear to emerge from the preceding chapters of this report. They can each be formulated in terms of distinctions between two aspects of the system.

- o A common and apparently fruitful approach to hypermedia systems is to distinguish the content from the hyperstructure. Standards work clearly distinguishes between these concepts, with standards such as MPEG, JPEG, G.72x, etc, for content; and HyTime or MHEG for structure. Currently-deployed systems also make this distinction, most obviously in Gopher, where the structure/content split maps onto the server filesystem's directory/file split. In a similar way, the ability to maintain hyperlink information separately from data is perceived in hypermedia research circles as a "good thing". Research systems such as Microcosm and Hyper-G do this, and HyTime with its ilink element also supports it. WWW does not support this, but requires link anchors to be edited into source data. There are problems with this approach, however - see the section on Microcosm for details.

- o A useful approach to content is to distinguish the media type from the media encoding. The MIME standard (used by HTTP2) illustrates how this can be done, and Gopher+ employs a similar system.

- o The distinction between data and protocol is also important for some systems. WWW for instance has clearly separate protocol (HTTP) and data (HTML) specifications. However, Gopher+ is specified without making this distinction. (The original Gopher system is very simple and arguably has no need for such separation.)

The most significant mismatches between the capabilities of currentlydeployed systems and user requirements are in the areas of presentation and quality of service. Adding flexibility in presentation capabilities to WWW or Gopher should be possible without any major change to the protocols (although it may require changes to data formats). Such capabilities could result from the progress towards greater integration of media types presaged above. However, improving QOS is significantly more difficult, as it may require changes at a more fundamental level. The following section outlines some possible solutions to this problem.

6.2. Quality of Service

Meeting the responsiveness requirement is certainly the key factor for the acceptance of networked multimedia information systems in the user community. To reiterate the requirement given in a previous section:

- For simple actions such as "next page", tolerable delays are of the order of 0.2s.
- For more complex actions such as "search for documents containing this word", then a tolerable delay is of the order of 2s.
- Users tend to give up waiting for a response after about 20s.

There are several methods which may alleviate the problem of poor responsiveness (or cause the user to revise his or her expectations of responsiveness!), some of which are described below.

1. Give clues that fetching a particular item might be time-consuming - simply quoting the size (and/or location) may be sufficient. WAIS and some Gopher clients already quote the size.
2. Display a "progress" indicator while fetching data.
3. Allow the user to interact with other, previously fetched information while waiting for data to be retrieved. The inability to do this is an annoying limitation of XMosaic. It can be difficult to implement, except on a multi-threaded operating system such as OS/2 or Windows NT.
4. Allow several fetches to be performed in parallel. Again, multithreading support makes this easier. This technique is less likely to be useful if all the nodes being requested come from the same server.
5. Pre-fetch information which the client software believes the user will wish to see next. This requires some "hints" in the data about which nodes might be good candidates for pre-fetching.
6. Cache information locally. The use of Universal Resource Numbers (see the section on WWW) is relevant for managing this.

7. Where multiple copies of the same information are held in different network locations, fetch the "nearest" copy. This is sometimes known as "anycasting", and is a more general case of local caching. The proposed URN-to-URL resolution service [26] could be used to support this.

8. When retrieving a document, the client should be able to display the first part of the document to the user. The user can then start to read the document while the system is still downloading it. Alternatively, the user may decide that the document is not relevant and abort the retrieval.

9. Offer multiple views of image or video data at different resolutions and therefore sizes. This enables the user to select a balance between speed of retrieval and data quality. Gopher+ and HTML2 both support this.

10. Future high-speed networks and protocols (ATM, RTP) will allow real-time display of isochronous data. Information systems should be able to take advantage of this.

A useful description of the problem is given in [27]. This paper rightly contends that the view, held by many hypermedia researchers and implementors, that the network is simply a transparent data highway which needs no special consideration in application design, is wrong. It is argued that:

> "the very same structural characteristics that may make a multimedia document appealing to the end user are the characteristics that are extremely helpful during dynamic network performance optimisation".

This is a particularly relevant statement considered in the light of suggestion 5 above.

6.3. Recommended Further Work

To meet the needs of applications such as those described in section 2.1, the community must seek where possible to adapt and enhance existing tools, not to build new ones. There is now an opportunity for RARE to stimulate and encourage this process of adaptation and enhancement, and the following subsections outline a strategy for this.

Selecting a System

In order to have the greatest effect, RARE should concentrate its efforts on only one of the existing tools. Candidate technologies are those already outlined: Gopher, WWW, WAIS, Hyper-G, Microcosm and AthenaMuse 2.

It is recommended that RARE should select the World-Wide Web to concentrate its efforts on. The reasons for this decision are as follows.

- Flexibility. The rich yet straightforward design of WWW, with its clearly separable components (HTML, URL and HTTP), means that it is a very flexible basis on which to develop distributed multimedia applications.

- Existing efforts. The WWW implementor community is already discussing and designing extensions to HTML (HTML2), intended (among other things) to support multimedia. There is clearly much interest in this area, and RARE efforts could complement existing work.

- Hyperlinks. A clear requirement of many applications is the availability of hyperlinking, which WWW supports well.

- Integrated solution. Because WAIS, Gopher and Hyper-G (as well as anonymous FTP servers) may all be accessed from Web clients, WWW serves as an important integrating tool for information services. It is important that distributed multimedia applications, which require extensive support in the client software, should be based on a technology "close to" such integrated clients.

- Penetration and growth. Although Gopher far surpasses WWW in the number of servers available, the rate of growth in WWW usage is greater than that of Gopher. There is an increasing realisation in the community that Gopher is over-simplistic for many purposes, and a corresponding increase in interest in WWW.

- Attention to QOS issues. There is already an awareness in the WWW community of the need for achieving an appropriate QOS, and a mechanism has already been proposed in HTTP2 to alleviate the problem.

- Standardisation. The WWW team is taking standardisation of the existing WWW system components seriously. The URL format has already been published as an Internet draft (and has been adopted as an important component of the proposed Internet integrated information infrastructure), and the current version of HTML is about to follow suit. The use of SGML as the basis of HTML complies with the perceived importance of SGML for hypermedia in general (and also fits in with RARE's approach of adopting appropriate open standards).

- o Software status. CERN has recently placed the WWW code developed by it into the public domain. This is unlike all the other candidate technologies, which all have restrictions on who can do what with the code. In the case of Gopher, these restrictions are already causing some commercial users to look at other options.

WWW has two significant disadvantages, both of which are being alleviated:

- o Restricted choice of client software. At present, Apple Macintosh and PC/MS Windows clients are available in beta form only. By contrast, there are more than one well-tested Gopher clients available for these platforms.

  However, other WWW clients for the Mac and MS Windows are in the pipeline.

- o There is a perception in the community that making information available over HTTP is difficult, and that it must be put into HTML.

  However, it is possible to put plain-text, non-HTML documents onto the Web. Such documents of course cannot contain links.

  Furthermore, WYSIWYG HTML text editors are available, to ease the pain of writing HTML.

The main disadvantages of the other systems are:

- o Gopher is designed for simplicity, and therefore lacks the flexibility of WWW. In particular its structure is too inflexibly hierarchical and it does not have hyperlinks. Its main advantage is its very heavy penetration. However, because of the WWW approach to accessing data using other protocols, all of gopherspace is part of the Web. Any Web client should be able to be a gopher client too.

It is neither envisaged that Gopher will go away, nor that it won't be used for multimedia data. However, Gopher is unlikely to be used for more sophisticated multimedia applications such as academic publishing, interactive multimedia databases and CAL, because of the above-mentioned limitations.

- o WAIS is a specialised tool, and will certainly form part of the overall solution, particularly for database-type applications. It is not a general solution for distributed hypermedia applications.

- o AthenaMuse 2 is commercially-oriented: it is clear that academic and research users will have to pay to use the software. Its level of use is thus very unlikely to be as great as publiclyavailable systems such as WWW. Moreover, it does not support all the required platforms.

- o Microcosm network support is still in early stages, limited at present to the PC/Windows platform. If it can be shown to perform adequately over a network, if it is capable of scaling to global levels, and if the advantages of maintaining link information separately from documents are found clearly to outweigh the consequent difficulties, it may become important in the future. Microcosm's authors need to ensure that the commercialisation of Microcosm does not hinder its adoption by the academic community.

- o Hyper-G is more difficult to dismiss. It is still in a relatively early stage of development, but appears to have many of the necessary features. Its main disadvantages are: (a) the lack of penetration outside the University of Graz - the author is aware of only one other site using it; and (b) it is currently limited to UNIX only. The author believes that, given WWW's head start in terms of deployment, and the current progress in adding multimedia facilities to it, WWW stands a much better chance than Hyper-G of being accepted as the de facto standard for distributed multimedia applications on the Internet.

Directions for RARE

Earlier in this report, it was noted that the most important areas where effort was needed were (a) provision of facilities for the integrated presentation of multimedia data (including synchronisation issues); and (b) ensuring adequate responsiveness.

Bearing this in mind, it is recommended that RARE should invite proposals and (subject to funding being available) subsequently commission work to:

1. Develop conversion tools from commercial authoring packages to WWW, and establish authoring guidelines for authors who wish to use the conversion tools. This is a significant and high-profile development aimed at enabling sophisticated multimedia applications to run over the network. (Authoring guidelines will be necessary to enable authors to fit in with the Web's way of doing things, and to document features of the authoring package which should be avoided because of conversion difficulties.)

2. Implement and evaluate the most promising ways of overcoming the QOS problem. This is an essential task without which interactive distributed multimedia applications cannot become a reality. Some possibilities have already been outlined in the preceding chapter.

3. Implement a specific user project using these tools, in order to validate that the facilities being developed are truly relevant to actual user requirements. It may be that partner funding from the selected user project would be appropriate.

4. Use the experience gained from 1, 2 and 3 to inform and influence the further development of HTML2 and HTTP2 to ensure that they provide the required facilities.

5. Contribute to the development of the WWW clients (particularly the Apple Macintosh and PC/MS Windows clients) in terms of their multimedia data handling facilities.

Although it is strictly speaking outside the remit of this report (since it is not specifically concerned with multimedia data), it is noted that the rapid growth of WWW may in the future lead to problems through the implementation of multiple, uncoordinated and mutually incompatible add-on features. To guard against this trend, it may be appropriate for RARE, in coordination with CERN and other interested parties such as NCSA, to:

6. Encourage the formation of a consortium to coordinate WWW technical development (protocol enhancements, etc).

7. References

[1] "A Survey of Distributed Multimedia Research, Standards and Products", ed. C. Adie, January 1993 (RARE Technical Report 5). URL=ftp://ftp.ed.ac.uk/pub/mmsurvey/

[2] "The Dexter Hypertext Reference Model", F. Halasz and M. Schwartz, NIST Hypertext Standardisation Workshop, January 1990.

[3] "Response Time and Display Rate in Human Performance with Computers", B. Shneiderman, Comp. Surveys 16, 1984.

[4] "Gopher+: Proposed Enhancements to the Internet Gopher Protocol", B. Alberti, F. Anklesaria, P. Linder, M. McCahill, D. Torrey, Summer 1992. URL=gopher://boombox.micro.umn.edu:70/11/gopher/gop her_protocol/Gopher%2b

[5] "WAIS Interface Protocol", F. Davies, B. Kahle, H. Morris, J. Salem, T. Shen, R. Wang, J. Sui and M. Grinbaum, April 1990. URL=ftp://quake.think.com/wais/doc/protspec.txt

[6] "Uniform Resource Locators", T. Berners-Lee, March 1993. URL=ftp://info.cern.ch/pub/ietf/url4.ps

[7] "The HTTP Protocol as Implemented in W3", T. Berners-Lee, January 1992. URL=ftp://info.cern.ch/pub/www/doc/http.txt

[8] "Protocol for the Retrieval and Manipulation of Textual and Hypermedia Information", T. Berners-Lee, 1993. URL=ftp://info.cern.ch/pub/www/doc/httpspec.ps

[9] "Hypertext Markup Language (HTML)", T Berners-Lee, March 1993. URL=ftp://info.cern.ch/pub/www/doc/html-spec.ps

[10] "Hyper-G: A Universal Hypermedia System", F. Kappe and N. Sherbakov, March 1992. URL=ftp://iicm.tu-graz.ac.at/pub/HyperG/doc/report333.txt.Z

[11] "Towards an Integrated Information Environment with Open Hypermedia Systems", H. Davis, W. Hall, I. Heath, G. Hill, Proceedings of the ACM Conference on Hypertext, Milan 1992, p181-190.

[12] "The AthenaMuse 2 Functional Specification", L. Bolduc, J. Culbert T. Harada, J. Harward, E. Schlusselberg, May 1992. URL=ftp://ceci.mit.edu/pub/AM2/funcspec.txt.Z

[13] "Research and Technology Development in Advanced Communications Technologies in Europe: RACE '92", CEC, March 1992. Available from: raco@postman.dg13.cec.be

[14] "Esprit Programme Synopses", CEC, October 1992. In seven volumes. Available from esprit_order_mailbox@eurokom.ie

[15] "CMIFed: A Presentation Environment for Portable Hypermedia Documents", G. van Rossum, J. Jansen, K. S. Mullender, D. C. A. Bulterman, Amsterdam 1993 (also presented at ACM Multimedia 93 conference). URL=ftp://ftp.cwi.nl/pub/CWIreports/CST/CSR9305.ps.Z

[16] "The Amsterdam Hypermedia Model: extending hypertext to support real multimedia", L. Hardman, D. C. A. Bulterman, G. van Rossum, Amsterdam 1993 URL=ftp://ftp.cwi.nl/pub/CWIreports/CST/CSR9306.ps.Z

[17] "Deja-Vu Distributed Hypermedia Application Framework", A. Eliens. URL=ftp://ftp.cs.vu.nl/eliens/Deja-Vu-proposal.ps

[18] "Bento Specification", J. Harris and I. Ruben, Apple Computer Inc, August 1992. URL=ftp://ftp.apple.com/apple/standards/Bento_1.0d4.1

[19] "Davenport Advisory Standard for Hypermedia (DASH), Module I: Standard Open Formal Architecture for Browsable Hypermedia Documents (SOFABED)", ed S. R. Newcomb and V. T. Newcomb. URL=ftp://sgml1.ex.ac.uk/davenport/sofabed.0.9.6.ps.Z

[20] Article in comp.text.sgml newsgroup, 24 May 1993, by Eliot Kimber (drmacro@vnet.ibm.com). URL=ftp://ftp.ifi.uio.no/SGML/comp.text.sgml/by.msg id/19930524.152345.29@almaden.ibm.com

[21] "Emerging Hypermedia Standards" B. Markey, Multimedia for Now and the Future (Usenix Conference Proceedings), June 1991.

[22] "Initial Draft PREMO (Presentation Environment for Multimedia Objects", ISO/IEC JTC1/SC24 N847, November 1992.

[23] "Recommended Practices for Multimedia Portability", Release 1.1 October 1990, Interactive Multimedia Association, 3 Church Circle, Suite 800, Annapolis, MD 21401-1993, USA.

[24] "Recommended Practices for Enhancing Digital Audio Compatability in Multimedia Systems", Release 3.00 1992, Interactive Multimedia Association, 3 Church Circle, Suite 800, Annapolis, MD 21401-1993, USA.

[25] "RIFF Tagged File Format", Microsoft Inc, 1992.

[26] "A Vision of an Integrated Internet Information Service", C. Weider and P. Deutsch, March 1993, Work in Progress.

[27] "Delivering Interactive Multimedia Documents over Networks", S. Loeb, IEEE Communications Magazine, May 1992.

[28] "A Status Report on Networked Information Retrieval: Tools and Groups", ed. J. Foster, G. Brett and P. Deutsch, March 1993. URL=ftp://mailbase.ac.uk/pub/nir/nir.status.report

8. Security Considerations

Security issues are not discussed in this memo.

9. Author's Address

Chris Adie
Edinburgh University Computing Service
University Library
George Square
Edinburgh EH8 9LJ
United Kingdom

Phone: +44 31 650 3363
Fax: +44 31 662 4809
EMail: C.J.Adie@edinburgh.ac.uk