Media Integration Issues within Open Hypermedia Systems

Hugh Davis, Wendy Hall and Ian Heath

The Image and Media Laboratory
The Department of Electronics and Computer Science
The University of Southampton
Southampton, SO9 5NH
Contact: Hugh Davis (hcd@uk.ac.soton.ecs) Tel. 0703 593669

Abstract

Multimedia facilities are now becoming common-place on desk top PC systems. In order for such facilities to be of use to users it is important that software seamlessly integrates the different media types. Hypermedia is a technology that can enable users to browse through a corpus of multimedia information, and to make and follow links within this information. At the University of Southampton we have developed an open hypermedia system known as Microcosm, which provides a link service across a range of applications and media types.

This paper describes the facilities available when dealing with textual information within Microcosm, and then examines the key issues which limit the extent to which these facilities may be extended to all media types.

1. Introduction

Hypermedia is a technology that allows users to browse through large bodies of multimedia information by a range of link-following navigational techniques, and to create new links in the material so that a network of conceptual links grows and may be maintained for later use.

Typical closed hypermedia systems hold the information about the links as mark-up within the data itself in much the same way that a word processor usually holds its formatting information as hidden mark-up within the textual data. The problem with this approach is that one has to commit ones data to the hypermedia program: once link data has been embedded in the user data, it will no longer be possible to view or edit the data using the original application that created it. Consequently such hypermedia systems tend to deal with static data, and thus have limited application areas.

The latest generation of hypermedia systems hold the information about links separately. This approach has the advantages that:

- It is possible to have links that have anchors in read only material, such as CD-ROM or videodisc.
- It is possible to have links into data, without putting mark-up into that data. This feature is important as it allows links to be made into and out of data that is created and displayed by a third party applications, removing the necessity for the hypermedia designer to duplicate their functionality.
- It is possible to manipulate and process the complete set of links. This is difficult if the link information is distributed over all the data files.
- It is possible to have multiple sets of links covering the same set of data. These different sets may be installed, joined and removed as required, making it possible to have different user views of a given data set.

One such hypermedia system is Microcosm (Davis et al., 1992, Fountain et al., 1990) which has been developed at the University of Southampton, and is currently in use in a range of application areas, including delivery of resource based educational materials, technical documentation and geographic/urban information systems. This paper describes the features of this system, and explores the issues that limit the ability to apply all the functionality to different media types.

2. The Microcosm Open Hypermedia System

Microcosm, which is currently implemented on the MS-Windows 3 platform, consists of a number of autonomous processes which communicate with each other by a message passing system. No information about links is held in the document data files in the form of mark-up. All data files remain in the native format of the application that created them. Instead, all link information is held in link databases, which hold details of the source anchor (if there is one), the destination anchor and any other attributes such as the link description.

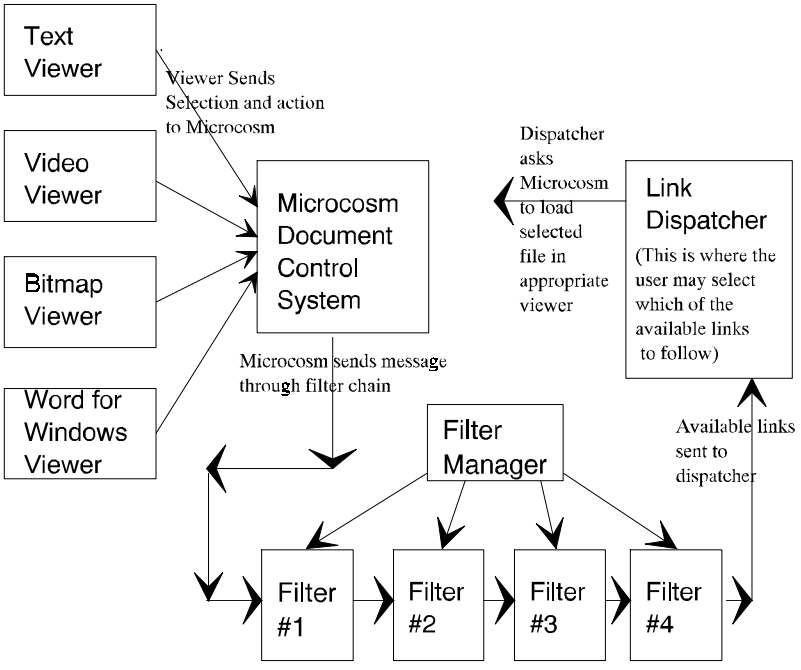

Figure 1: *The Microcosm Model*

Microcosm allows a number of different actions to be taken on any selected item of interest, so consequently use of the system involves more than simply clicking on buttons to follow links. In Microcosm the user selects the item of interest (e.g. a piece of text) and then chooses an action to take. We may see this as selecting an object then sending it a message. A button in Microcosm is simply a binding of a specific selection and a particular action. A particular feature of Microcosm is the ability to generalise source anchors. In most hypertext systems the source anchor for any link is fixed at a particular point in the text. In Microcosm it is possible for the author to specify three levels of generality of link sources.

1) **The generic link**. The user will be able to follow the link after selecting the given anchor at *any* point in any document.

2) **The local link**. The user will be able to follow the link after selecting the given anchor at any point in the current document.

3) **The specific link**. The user will be able to follow the link only after selecting the anchor at a specific location in the current document. Specific links may be made into buttons.

Generic links are of considerable benefit to the author in that a new document may be created and immediately have access to all the generic links that have been defined for the system.

The basic Microcosm processes are viewers and filters.

2.1. Viewers

Viewers are programs which allow the user to view a document in its native format. Included with Microcosm are viewers for an number of common formats, including ASCII text, rich text (RTF), Windows bitmaps, Windows meta files, digital video, WAV, MIDI and CD audio.

The task of the viewer is to allow the user to peruse the document, to make selections and to choose actions. Typical actions are *follow link, make link* and *complete link* (where links may be to processes as well as to documents). The actions themselves are not effected by the viewer. The viewer is responsible for binding the information into a message, which is sent on to the filter chain where it will look for one or more processes that can satisfy this request. Any Windows application might be used as a viewer, with the proviso that it is possible to select objects, and either communicate an object to the Dynamic Data Exchange (DDE) or copy it to the clipboard.

A major strength of Microcosm is its ability to integrate other applications. In fact Microcosm may be seen as an umbrella environment, allowing the user to make links from documents in one application package to documents in another application package.

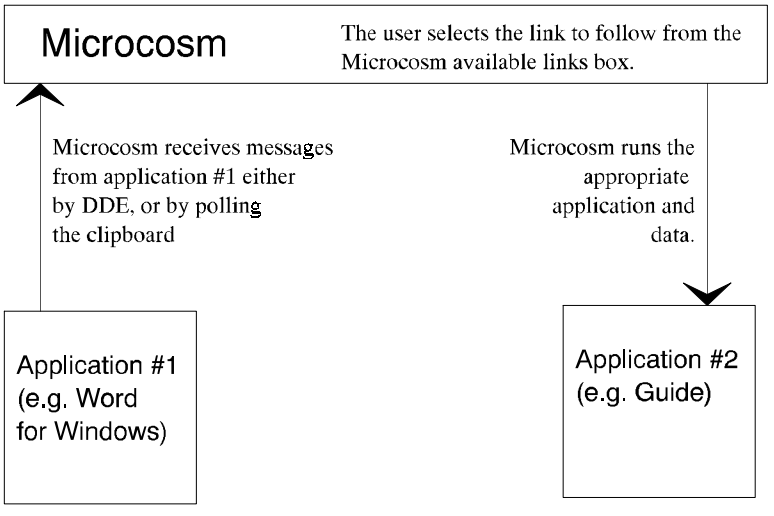

Figure 2: *Microcosm as an environment for integrating applications and tools.*

2.2. Filters

Filters are processes which are responsible for receiving messages, taking any appropriate actions, and then handing the message on to the next filter in the chain. The actions that filters take will be of the nature of changing the message, or adding or removing messages. The order that the filters appear in the chain is under user control, and may be dynamically re-ordered and filters may be installed and removed. The filters are a particularly important aspect of Microcosm as they provide the hypermedia functionality. A range of filters have been produced, which have functions such as link identification, link making, navigational aid, network communication, a query command line and information retrieval. Two important types of filter are:

Link Databases
Link Databases hold all the information referring to links. More than one database may be installed at a time. When new links are created they are inserted into the first link database in the chain. This makes it possible to have a concept of public and private databases. The public database may contain all the links made, say, by the original author, and private databases may contain links made by individual users. This ensures that private annotations and links do not become absorbed into the view of the system, as seen by other users.

Compute Links
Sometimes no links have been defined for a particular subject. On these occasions it is desirable to offer the user some further assistance. Microcosm has a facility that allows a user to batch a set of text files and to index these documents (Li et al., 1992). Once this indexing has been done a block of text may be selected and the action, compute link, may be chosen. The system will very rapidly return a number of other documents within the system that have a similar vocabulary to the selected block, in the order of best match. Clearly this filter is only able to identify links to text documents.

3. Multimedia Issues

In designing Microcosm, we have attempted to maintain a consistent interface to all different media types. Generic and computed link types are currently only available for links made from text documents for obvious reasons, but we are working to extend these link types to other media. This work is discussed later in this section. How links are displayed as buttons clearly varies from media type to media type. In text, buttons are hi-lighted dynamically as the text is displayed. If the user asks to "show links" then the local and generic links anchors are also hi-lighted. There are a number of user-interface issues involved here. Some users would prefer all links to be hi-lighted automatically. This is very processor intensive and can lead to "link-overload" if the linkbases are very large. The same is true for non-text documents. The 'show link' facility is available for all media types and in the case of in non-text documents the buttons in that document are displayed to the user. This is not consistent with the text documents, where buttons are automatically displayed but too many buttons can clutter a picture or video sequence and users seem to prefer to have the option of whether to display the buttons or not. Displaying buttons in a sound sequence is a non-trivial interface problem as is the whole issue of displaying sounds sequences in a meaningful way. We have produced a prototype 'sound viewer' to experiment with this.

Consistency across interfaces to different media types is very important in providing the user with an integrated information environment. This will be particularly important when the information is created and used by groups of people over a network. We have a large Teaching and Learning Technology Project (TLTP) running at Southampton to create a central database of multimedia resource material to be delivered across the campus network for use by staff and students. Microcosm is the software platform that will provide the mechanism for the delivery of this material, integrated with various database and information retrieval technologies. Issues of media integration are therefore central to our development plan. Our approach to some of these issues is described below.

3.1. Hardware platforms, transfer rate and storage size.

A considerable advantage of using MS-Windows as an operating environment is the fact that it provides a device independent interface, which means that it is possible to produce code that will run on a range of different hardware configurations. However, as yet there is no real consensus (in spite of various attempts such as Microsoft's MPC standard) as to what configuration represents the true multimedia platform: to a certain extent this is inevitable because as technology moves forward the limits of what is possible and affordable expand, so any fixed definition is doomed to become quickly outdated (as did the original IBM specification based upon 286 machines). In practice the machines on which Microcosm data sets are most

frequently delivered tend to be 386 PC's with limited screen resolutions and colours, hard disks in the order of 100MB, and with no special multimedia hardware such as CD player, MIDI player or video hardware. This is in marked contrast to the machines for which we design our systems and on which most data sets are implemented which tend to be 486 PC's with state of the art screen technology and multimedia devices. The consequence of this mismatch is that we must design our data sets in such a way that they can operate in a limited environment, while offering added value where appropriate hardware is available.

A specific issue concerns the problem of storing and transferring files, particularly sound and video. In the "ideal" environment all users would sit at terminals which were connected to a CD player and a videodisc player. However cost prohibits such a solution, and anyway in an open resource based system, the users would spend all their time swapping disks rather than moving effortlessly from one piece of data to another as preferred. Some projects have attempted to circumvent this problem by installing "juke-boxes" and broad band cable networks for delivering such media (e.g. Applebaum, 1989), but these solutions are costly, and anyway not commonly available. For these reasons we have moved towards the use of software digital video and sound. Such files tend to be very large by current norms. E.g. one minute of compressed 320 by 200 video might typically require 3 megabytes of storage and one minute of reasonable quality digitised sound might require 1 megabyte. This presents problems since such files quickly become too large to be held in the memory of current machines so they must be played from secondary storage. When the secondary storage is the local hard disk or CD-ROM, the delivery rate will vary depending upon the transfer rate available from the disk, so it is not possible to define a standard quality, and when the secondary storage is a fileserver, transfer rates will depend upon network loading.

Wherever large data files are stored there will be a time delay in following a link to such data. We have used the idea of *Micons* which are abstracts of video files, as introduced in Brøndmo and Davenport, 1990. Whenever a link is followed to a destination which is video, the user is shown a small bitmap picture, or a few frames of video playing in a loop in such a way as to convey sufficient information about the intended video, so that the user is able to decide whether the cost of following the link through is appropriate. This also provides a minimum fallback situation in the case where the intended link is to a CD-ROM or videodisc which is unavailable. In the case of audio destination files the best we have been able to do is to offer textual abstracts to describe the intended file.

3.2. Indexing multimedia items

A well reported problem in the multimedia database world is that of indexing data items (O'Docherty & Daskalakis, 1991., Gevers and Smeulders, 1991., Orlandic, 1992., Hardman et al., 1993.). How does one allow the user to query the database? Apart from the normal link following mechanisms, Microcosm has three methods of querying the resource base:

1. Keywords may be associated with all data items, and then all the normal Boolean logic operations may be applied to retrieve sets of documents that meet particular criteria. This method depends upon the skill and care with which keywords are attached to documents.

2. All documents are filed in directories, and then given textual descriptions. In just the same way as one routinely navigates any hierarchical file system, it is possible to search for files in Microcosm. This method depends upon the author providing an appropriate description and upon the file structure being carefully pre-organised.

3. In the case of text documents only, it is possible to use the computed linker. This is a text retrieval system, described earlier, which enables the user to query the system based upon statistical similarity between given documents and the query: it is a highly successful method but is not extensible to other types of media.

A further problem involves extending generic links to work with source anchors in media other than text. A text string makes a sensible anchor for a generic link, as a given string is identical from whatever place it is selected, and by using a thesaurus it is possible to follow links from synonyms. However what would be the equivalent of a generic link from say a picture or a video? Ideally what one would wish is to be able to say, for example, whenever a user clicks on a picture of a particular person, follow a link to a biography of that person. However, such a scheme would require the software to have much greater intelligence than is currently achieved.

Various solutions are being investigated within Microcosm. One solution is to associate documents together in "compound documents". Such documents consist of text automatically linked to other media, in such a way that whenever a text file is offered as a destination of a link or any other search, the other linked documents will also be offered as well. Thus, for example, one could discover a biography of a particular character by following a computed link, and then be offered video, pictures and audio recordings of this character at the same time. We are also starting some research in the area of generic links within pictures, by investigating the possibility of identifying images from within a limited domain within a picture. As an example, ordnance survey maps are created from a finite set of regular symbols, and it should be possible to apply image processing techniques in real time to identify the selected symbol and to follow a generic link to information about that symbol.

Another research area involves buttons in moving video. At present the author creates an active area over the first frame containing the object on which the button is to be placed, then plays the video, while manually dragging the active area along on top of the object. The details of the active area are then saved so that they can be replayed with the video. However, image tracking algorithms are now quite sophisticated (Dobie & Lewis, 1992) and it should be possible to provide batch tools for the author to create moving buttons automatically.

3.3. Synchronisation of Multimedia Items.

In a closed system where a single program is used to view all formats of multimedia data it should be possible to control the playback speed of data that has a temporal dimension such as audio or video, in such a way that they are synchronised. For example Microsoft's AVI format for Windows allows the simultaneous playback of digitised video files interleaved with sound that was captured at the same time. However, if the files were captured at separate times it will not be possible to synchronise them, as there is no mechanism to flag the points at which synchronisation should occur, and there is no guarantee, even if the playbacks are started simultaneously, that they will progress at the original rates.

There has been much recent research in the area of specifying and delivering synchronised multimedia documents (Bulterman et al., 1991., Buchanan and Zellweger, 1992., HyTime, 1992., Hardman et al., 1993., Ramanathan & Venkat Rangan, 1993). However, the bottom line is that synchronisation between separate processes can only be achieved if appropriate services are provided at operating system level. Currently the Windows Media Control Interface (MCI) does not provide such facilities, unless a single hub program is given control of all media: this seems inconsistent with Microcosm's open approach.

The ability to provide synchronisation between separate processes in an open system is clearly essential. Examples that we have been involved with include

- Playing alternative commentaries for a piece of video.
- Playing Music beside musical score (preferably with a "now point" in the music).
- Displaying help text at various points in a user driven simulation.

It has been possible to achieve a rough and ready solution to these problems, but proper synchronisation facilities would be preferred. The examples above indicate the range of media types which might need to

be synchronised: the combinations are endless, and a solution which involves a single application to deliver all the different media is necessarily limited.

3.4. Multimedia Data Standards

In the ideal world it would be possible to move around global networks in search of multimedia data, and then to make links to that data. However, currently the world of data standards is very confusing to the user. There are dozens of different formats for representing pictures. Even ASCII text is represented differently on Unix machines and PC's. Multimedia standards do not seem to be developing any more consistently; there are already a number of formats for representing digitised sound and video. From our point of view as hypermedia tool providers, the current situation is made even more complex as we not only require a standard for content interchange, but also require a standard for hypermedia structure interchange. Such standards are beginning to emerge (HyTime, 1992., MHEG 1992., HyperODA 1992), but as yet these standards are little more than paper implementations as no engines have yet emerged which we can use for hypermedia interchange.

4. Conclusions

Clearly issues of multimedia integration are central to the development of successful multimedia information systems. Our philosophy is to provide access to an integrated multimedia environment via an open hypermedia link service so that users have a consistent interface to multimedia information which they can access from within any information processing environment. In this way it is possible to harness the full power of databases, spreadsheets, text-processors etc. whilst providing the user with integrated access to a common multimedia resource base. This approach naturally extends to a distributed environment and will support the creation and use of shared resource bases. The facilities provided for integrating different media types are however still very limited in the sense that what we can do with text, we cannot yet do with non-text data types. This paper has described some of our solutions to these problems and some of the issues that still have to be resolved. There is no doubt however, that even the limited facilities that we can offer users today for interacting with non-text information greatly enhances their working environment, and solutions to the problems of media integration will be one of the main driving forces behind the software developments of the 1990's.

References

Applebaum D.I. (1989). Galatea. Massachusetts Institute of Technology Media Laboratory Technical Report. MIT.

Brøndmo, H.P. & Davenport, G. (1990). Creating and Viewing the Elastic Charles - a Hypermedia Journal. *in* McAleese, R & Green, C. *eds*. *Hypertext, State of the Art*. Intellect Ltd.

Buchanan, M.C. & Zellweger, P.T. (1992). Specifying Temporal Behavior in Hypermedia Documents. *In:* D. Lucarella, J. Nanard, M. Nanard, P. Paolini. *eds. The Proceedings of the ACM Conference on Hypertext, ECHT '92 Milano*, ACM. 262-271.

Bulterman, D.C.A., van Rossum, G. & van Liere, R. (1991). A structure for Transportable, Dynamic Multimedia Documents. *in*: *The Proceedings of USENIX* 1991.

Davis, H.C., Hall, W., Heath, I., Hill, G. & Wilkins, R. (1992). Towards an Integrated Information Environment with Open Hypermedia Systems. *In:* D. Lucarella, J. Nanard, M. Nanard, P. Paolini. *eds. The Proceedings of the ACM Conference on Hypertext, ECHT '92 Milano*, ACM. 181-190.

Dobie, M.R & Lewis, P.H. (1992). Object Tracking in Multimedia Systems. in: *The Proceedings of the 4th IEE Conference on Image Processing and its Applications, Maastricht.* pp 41-44

Fountain, A.M., Hall, W., Heath, I. & Davis, H.C. (1990). MICROCOSM: An Open Model for Hypermedia With Dynamic Linking, *in* A. Rizk, N. Streitz and J. Andre *eds*. *Hypertext: Concepts, Systems and Applications. The Proceedings of The European Conference on Hypertext, INRIA, France.* Cambridge University Press.

Gevers, T. & Smeulders, A.W.M. (1991). Indexing of Images by Pictorial Information. in: *IFIP WG 2.6, 2nd Working Conference on Visual Database Systems, Budapest Hungary.*

Hardman, L., Bulterman, D.C.A. & van Rossum, G. (1993). The Amsterdam Hypermedia Model: Extending Hypertext to Support Real Multimedia. *Hypermedia*. (In Press).

HyperODA (1992). ISO/IEC JTC1/SC18/WG3 N1898. HyperODA - a Working Draft for Extending ODA Standards to Support Hypermedia Applications

HyTime (1992). ISO/IEC 10744. Hypermedia/Time-based Structuring Language.

Li, Z., Davis, H.C. & Hall, W. (1992). Hypermedia Links and Information Retrieval. *The Proceedings of the 14th British Computer Society Research Colloquium on Information Retrieval*, Lancaster University.

MHEG (1992). ISO/IEC JTC1/SC29/WG12/NO26. The MHEG Standard and its Relation with the Multimedia and Hypermedia Area.

O'Docherty, M.H. & Daskalakis, C.N. (1991). Multimedia Information Systems - The Management and Semantic Retrieval of all Electronic Data Types. *The Computer Journal* 34(3), 225-238.

Orlandic, R. (1992). Problems of Content-Based Retrieval in Image Databases. in: *Proc. 3rd Symposium on "New Generation" Knowledge Engineering, IAKE '92, Washington D.C.*

Ramanathan, S. & Venkat Rangan, P. (1993). Feedback Techniques for Intra-Media Continuity and Inter-Media Synchronisation in Distributed Multimedia Systems. *Computer Journal* 36(1) 19-31.