Putting the Media into Hypermedia

N. D. Beitner, C. A. Goble†, W. Hall

Department of Electronics and Computer Science
University of Southampton
Southampton SO17 1BJ, UK
e-mail: {ndb88r, wh}@ecs.soton.ac.uk

†Department of Computer Science
University of Manchester
Oxford Road
Manchester M13 9PL, UK
e-mail:carole@cs.man.ac.uk

ABSTRACT

This paper discusses the issues involved in putting the media into a hypermedia system. The main argument of the paper is that to-date media representation and underlying link strategies have been too closely tied together in the move from hypertext to hypermedia. We argue that it is necessary to separate the issues of media from link structure, and we present a model which solves some of the problems of genuine media integration in a hypermedia system. At the same time the model provides support for the creation of links between data of different media types in a conceptually meaningful way.

The paper describes the design of Microcosm++, an object oriented extensible service-based architecture for building consistent integrated hypermedia systems. It is based on the Microcosm hypermedia system which was developed at Southampton. The current implementation of Microcosm++ demonstrates the flexibility of object-based services for making hypermedia more viable in a working environment. The approach described reduces authoring effort significantly while at the same time increasing the integrity of the link structures and providing a unified model for media integration.

KEYWORDS: Open hypermedia systems, Linking mechanisms, Multimedia, Object oriented services.

1. INTRODUCTION

As we have progressed from hypertext to hypermedia, the relationships and dependencies between different media types have not always been considered by researchers in the hypertext/hypermedia community. The tendency has been to take existing hypertext systems and extend them to include new media types without consideration of what it means to make links between them. This has come about in the rush to be the first to use multimedia information in traditional hypertext environments. The underlying structures are biased towards textual data, and there seems to be little consistency across media. Hypermedia systems have now developed to a level of sophistication where this issue can be freshly examined to develop a new generation of hypermedia systems in which multimedia information is truly integrated.

This paper discusses the issues involved in putting the media into a hypermedia system. The thrust of the paper is that to-date media representation and underlying link strategies have been too closely tied together in the move from hypertext to hypermedia. We argue that it is necessary to separate the issues of media from link structure, and we present a model which solves some of the problems of genuine media integration in a hypermedia system. At the same time this model provides scope for a new linking architecture that provides support for the creation of links between data of different media types in a conceptually meaningful way. Microcosm++ is an object oriented extensible service-based architecture for building consistent integrated hypermedia systems.

The paper begins with a short summary of related work and discusses the shortcomings of current models in respect to media integration, including our own hypermedia system Microcosm. It describes work we have undertaken within Microcosm which has resulted in ideas for a new unified model for media integration. This model is presented and discussed in the final section of the paper.

2. BACKGROUND

There are very few hypertext systems that do not claim also to be hypermedia systems. Usually this means that the original hypertext model has been extended to allow source anchors to be defined in certain limited media types and to allow the destination of a link to be a document of any media type. The latter is common, the former less so because of the inherent difficulties of creating and representing source anchors in media other than text. Node to node links are more common in multimedia systems because these problems are then simplified. When linking is attempted in media other than text it is usually through explicit point to point links which take little or no account of the media involved, and amount to little more than hot spots. For example the latest portable document structure from Adobe, Acrobat[1], demonstrates the difficulties and results in link structures whose anchors are simply region definitions with no idea of content.

In the current literature there is little discussion of what it involves to define a link from an object in a non-text document to objects in other documents, other than to say that there is an association between the objects identified. Few projects have considered the problems of authoring support for the creation of links between different media types, nor issues of support for users in effective navigation of an information space, particularly with respect to temporal media such as video and sound. In this section we discuss some of the projects that have attempted to address these issues.

2.1 Intermedia

The Intermedia hypermedia system[27] supports the creation of links between different media, and provides an interface that allows the user to navigate the resulting webs created. Intermedia is fundamentally a closed explicit authoring environment which supports a variety of media which have to be encoded in their own proprietary data structures. With the integration of temporal media, such as video and animations, they developed *active anchors*[23]. Intermedia active anchors do not refer to any temporal behaviour of an anchor within a dynamic document, rather they refer to the links that are bound to destination documents. Their primary consideration was the behaviour of documents once a user followed links to them. The active aspect of a link determines what should happen to the destination document when it is presented; if it is video or audio should it be played? An example they give is the possibility of recording some audio as a result of following a link, or initiating a database query. Such a link extension has no direct correlation to the extension of a hypertext system to encompass multimedia.

Although Intermedia puts much emphasis on providing a consistent interface to links and anchors across media, there are a number of limitations. It provides a limited anchor model for dealing with links from temporal media, buttons in temporal media are not similar to those seen in still images and can only represent a temporal segment of the whole image shown. In effect at the appropriate moment in a sequence the whole video window becomes a single anchor. When the scope of the link expires the viewing window ceases to be active. The context of such links is very restrictive and fails to allow the user to distinguish between the available links during an active temporal segment.

While Intermedia has a high level of functionality including many useful navigation tools, the system as a whole suffers from inconsistencies at the media integration level due to the initial lack of consideration for future multimedia support.

2.2 Elastic Tools

The Media Lab in MIT developed the Elastic Tools[4] to allow the easy development of multimedia interactive applications. The tools themselves do not form a hypermedia system, but allow authors to generate applications with hypermedia behaviour. However the system imposes a definite distinction between authors and users; a structure is authored using the tools and then passed over to users for browsing activities.

The project involved experimentation in new areas of media interaction, and developed ideas that if applied in real hypermedia systems would do much to support and complement existing models. *The Elastic Charles* project[5] is a demonstration of the 'elastic' tools that are discussed here. The aim was to build a hypermedia journal - "The Elastic Charles" - as a hypermedia edition of the hypothetical "American River Magazine". The magazine was authored by fifteen students and contained stories focusing on the history, ecology, recreation, and construction relating to the Charles River. Most of the stories had a strong video component which was augmented by text, graphics and sound.

The work on the Elastic Charles was the first to really consider the place of video in hypermedia and its impact. Where a hypertext system is completely user driven, the addition of dynamic media causes many problems with links and user response. Brøndmo used the concept of micons (motion, moving or movie icons) to represent links to video inside source documents.

The traditional icon is a pictorial representation of an entity. The micon works in much the same way except that it is animated. Brøndmo used micons to represent links to video. A small section of the destination video is digitised and played in a loop; it is used to indicate to the user that there is a link to video and also to provide some information about the sequence before it is viewed. This mechanism is more informative than a text description alone.

The problems of links from temporal media are tackled by generating a list of links while the temporal medium is being viewed: that way if a link is missed the user can follow it at a later date. Links to video pop up as micons at the appropriate moment, and when the link becomes invalid, in a temporal sense, it is removed. This model is superior to that of Intermedia, as it distinguishes between the different link anchors. Links do not have to be merged into one anchor, and users are able to select an anchor of interest when multiple anchors appear simultaneously. The use of micons to represent links to video allows users to make an informed choice about which links to follow. It is evident that this extension of icon representation is beneficial to a user interface within a multimedia system. At the time of the Elastic Charles project, Brøndmo only made use of micons to represent links from video to video: it is clear that there is wider scope for application. Micons do not allow authors to make links from particular entities within a temporal sequence so they do not correspond to the metaphor of link following from a text object used in text based systems.

In summary micons preview the destination document from the source document, improving user navigational behaviour. Their down side is that they interfere significantly with the source document and are restricted in the types of link and the number of destinations they can represent.

2.3 The Amsterdam Hypermedia Model

The Amsterdam Hypermedia Model[13] considers the problems of extending hypertext models to include multimedia information but concentrates on the issues involved in integrating time-based presentations that consist of a number of simultaneously active media items. The model is an extension of the Dexter hypertext reference model[12] developed to integrate time at a fundamental level in the model, thus leading to the definition of contexts for a link. They define the source context for a link as that part of a hypermedia presentation effected by following a link, and the destination context is that part of the presentation which is played on arriving at the destination of a link and extend these ideas in a subsequent paper[14]. A context for a link makes explicit which part of a presentation is affected when a link is followed from an anchor in the presentation. Given explicit source and destination contexts for a link, an author is able to state the desired presentation characteristics for following links, including whether the presentation currently playing should continue or be replaced.

This work is closely related to recent research in the area of specifying and delivering synchronised multimedia documents e.g. Buchanan[6] and Newcomb[21]. It is a very important issue in the development of multimedia systems, and does tackle the issues of extensible hypermedia in terms of integration of multimedia with different link strategies.

In this paper we are more concerned with techniques for link creation and link following between different multimedia documents rather than within composite documents.

In the next section of the paper we describe an open hypermedia system that has been developed at the University of Southampton and discuss ways in which we have experimented with integrating multimedia information into this system.

2.4 Microcosm

Microcosm is an open hypermedia system currently implemented under Microsoft Windows. Within Microcosm it is possible to browse through large bodies of multimedia information by following links from one place to another[7, 9]. In its simplest form, Microcosm can be considered as a link service which communicates with any number of autonomous processes (*viewers* and *filters*) via a message passing system. An important feature of the Microcosm link strategy is that no information about links or anchors is held in source or destination documents as mark-up. Instead, all link information is held in separate link databases. This makes it possible for processes to examine the complete link database as a separate item. In addition this allows us to make link anchors in documents that are held on read only media such as CD-ROM and videodisc as well as with third party proprietary document types which are handled by their own respective document viewers (Image files, Word Processor Documents, Spread Sheets, etc.).

Microcosm allows any number of customisable actions to be taken on any selected item of interest, resulting in a system that supports more than simply clicking on buttons to follow links. In Microcosm the user selects the item of interest and then

chooses an action to take. A button in Microcosm is simply a binding of a specific selection and a particular action. Each of the actions/functions is provided by a filter and new filters can be added to extend the systems functionality[15].

In addition to traditional explicit link authoring, the specification of active regions of a document, a particular feature of Microcosm is the facility for *implicit* anchor specifications for text, known as generic links. A generic link is a generalisation of a text anchor specification. Instead of an anchor being described using some form of co-ordinate system, the source anchor is encoded using the textual representation of the object bounded by the anchor. Depending on the constraints associated with the generic link any occurrence of the same text selection in other text documents automatically becomes an anchor for that link.

Explicit anchor definitions can be used for button definitions or hidden anchors. The former presents itself to the user and is easily used, the latter are not automatically shown. This invisible anchor does not interfere with the presentation of a document, but users are not easily aware of their existence. Generic links are not bound to a specific document, and by their unconstrained nature have this disadvantage. They allow a small number of links to be available from a large number of locations, but users must actively search for them.

Generic links allow the rapid incorporation of new material, since any new document immediately provides access to all those generic links whose anchors it contains. However, the new document does not become the destination of any links until explicitly specified as such by the author. The use of dynamic destination anchor resolution functions, like full text retrieval[18] or document attribute keying, allows predefined links to return newly added documents with only minimal extra authoring, and reduces the need to define more explicit or implicit links.

As far as interface behaviour and link structure is concerned, Microcosm provides exceptional explicit link support in many media including, text selections, regions in still images, dynamic moving buttons in video[2], and temporal anchors in sound documents[11]. It is our belief that in these terms Microcosm is one of the most consistent systems available.

Nevertheless, as with many other hypermedia systems, Microcosm was built upon a fundamentally text-oriented architecture. In its present form it presents a number of restrictive practices that impede the development of advanced link services across the range of available media.

While the document viewers allow multimedia interfaces, the navigation and authoring tools provided within the Microcosm framework have a predominantly textual look and feel. The underlying link representation and storage is also text based. For example, the link database encodes anchors in terms of text selection and offset, so all other representations must be converted to a textual representation which is encoded using these two terms. This leads to obvious restrictions when considering other media types.

While creative implementation can allow latitude within these storage requirements, the interface issues in the current model are not so easily circumvented.

During the process of improving the user interface, extending generic link functionality and the consideration of generic links in media other than text, it became evident that the architecture could no longer stand the constant 'creeping featurism' that has typified the development of hypermedia to date.

Consequently an object oriented support services approach has been developed to allow separation of hypermedia issues from the technical problems of providing specific support for media. The remainder of this paper discusses re-engineering of the Microcosm architecture to allow object oriented support services and an improved interface for media interaction.

3 MICROCOSM++ WITH OBJECT SUPPORT SERVICES

Microcosm is fundamentally a message passing architecture that allows applications to communicate requests, and others to perform actions as a result of those requests. On top of this architecture is a model for hypermedia which is implemented using the communications architecture. There are a number of components required to make up a hypermedia system using the Microcosm communication protocol.

- Document viewers allow users to visualise data and initiate requests through the Microcosm communications chain.
- Link databases hold information that describe anchors and resolutions, which are used to provide the structure that typifies hypermedia networks.
- Components of the action processing system (APS) are responsible for evaluating resolution specifications and producing the desired results of following links e.g. Starting up document viewers, initiating database queries, etc.
- Authoring tools are required for creating links and structure of the information network.
- Navigational tools are required to improve the users ability to effectively locate required material.

Under Microcosm++, object support services are a means for tools to describe functional behaviour for specific media without encoding the support within the service application. In cases where tools share media specific requirements object services act as a common resource service. Navigational tools may use common objects services to render anchors, while link databases use application specific media services for anchor indexing and searching.

In order to indicate the location of generic links in text documents, the link database must be aware of the media involved, be able to load the current document, and search through the document for valid occurrences of an anchor within it. One of the first tasks of object support services was to allow the link database, or linkbase, to compare the link specification index against a document and derive a set of "explicit" anchors from the generic list for the text viewer to show as buttons (Fig 2b). This removes the requirement for a user to actively select a block of text and query the link base for an anchor match. Once generic link specifications can be used to describe buttons their value as an authoring tool becomes much more significant.

Object services allow the linkbase to separate the techniques for matching generic link specifications from the basic linkbase implementation. Depending on the document being viewed an object service provider matches link specifications against the document and returns a set of buttons.

In order to provide such a service, an API for anchor specific functions is defined by the developer of the linkbase, and these are implemented in the object support services hierarchy (see Fig. 1). The functions are geared around providing the following services.

- Find buttons - Given a document return any explicit or implicit derived anchors that are applicable and should be presented as buttons.
- Follow link - Given a selection from within a document, return the set of resolutions that are bound to anchors that satisfy the selection.
- Show links - Similar to Follow link except that Follow link uses an exact match, where as show links looks for a match within the overall selection object.
- Define anchor - Defines some object based specification that will be used to locate an anchor within a document or a selection of a specific media type.

The orriginal version of Microcosm only has a generic anchor algorithm for text, algorithms such as those used in "Miyabi"[16, 22] for simple image matching can also be applied without any changes to the actual link database tool. Work is currently being undertaken to develop these ideas using the Microcosm architecture[26].

As well as using object services for providing extensible media support for tools to use, tools can also provide object services to cater for extension or replacement of shared components of the system. A link is the unification of a source anchor specification, which is link database specific, and some resolution which is APS component specific. The link creation tool needs to be able to cater for various anchor types, linkbase specific requirements and resolution specific requirements. As such the linker requires the object services hierarchy to provide the following support:

- General media support to allow the rendition of anchors during anchor selection.
- A link database specific object to define source anchor entities, which itself is also likely to make use of general media services.

- A resolution specific object server for actions supported by the APS, which also is likely to make use of the media services suite.

In the case of the link database, the object services hierarchy (OSH) is used to provide media specific support, the link generator also uses it to provide extensible linkbase and APS support services.

Fig 1. Shows how different aspects of Microcosm tools may be integrated in the OSH.

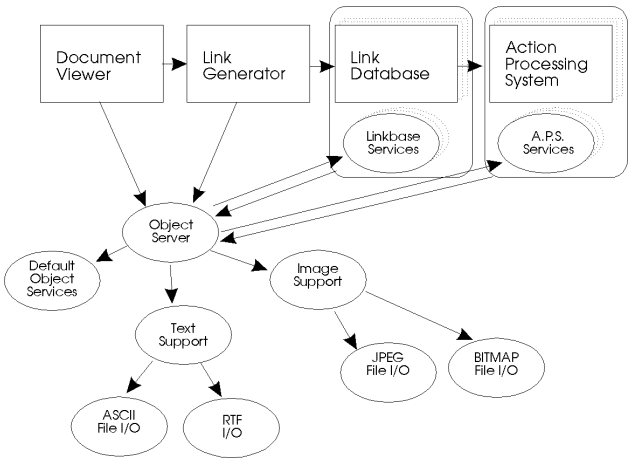

Fig 1. A typical object services hi.erarchy (OSH)

Document viewers (e.g. Fig. 2a & 2b) may use the objects services to render the document objects, thus allowing a single common application to handle rendition of a large number of specific data types of a basic media subset, such as general still image and video support. A viewer may also use them to encode selection objects for transfer through the Microcosm communication system.

Fig 2a. Arbitrary selection from a still image.

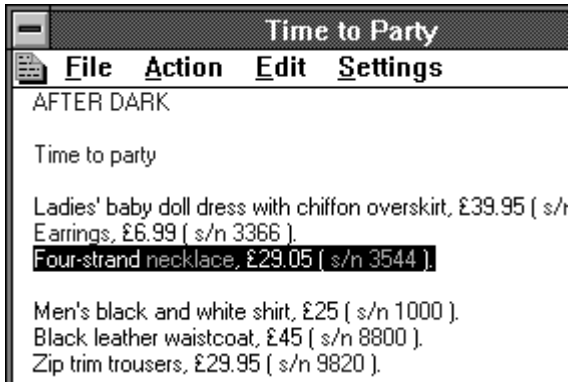

Fig 2b. Arbitrary selection from a text document.

The link generator (Fig. 3) uses the OSH to render anchors for consistent user interface behaviour, and also uses the OSH to generate the appropriate source anchor specifications for a given linkbase as well as to generate the resolution to be evaluated by an APS component when the link is eventually followed.

Figures 2a, 2b, 3 & 4 show the interface consistency that occurs as a result of having the OSH for rendition. Each of the services is dynamic and can be changed while the applications are running. When a new media rendition service is added, say for a third party application, the anchor objects can automatically be visualised in the navigation and authoring tools. An audio service provider is next to be developed, and without any extra work to the link generator, anchor visualisations for sound will be available.

Fig 3. Anchor collection in the link generator tool.

Finally the linkbase uses object services to provide anchor specification support for media. Interfaces for specific link database implementations are implemented at the OSH level and the linker uses those services to define the anchor specification that will eventually be stored in the link database. The linkbase service objects use the OSH to provide specific media support for the overall linkbase service object.

In a similar manner to the linker, the Microcosm++ available resolutions interface (Fig. 4) presents a list of destination documents as a result of a link following action or selection action process. Whereas the linker attempts to render the exact selection made in defining an anchor, the available resolutions interface provides document abstraction.

4 DOCUMENT ABSTRACTION

Document abstraction is a further function of the OSH, and makes use of other media object services functions. There are three types of document abstract.

- Link abstractions are the definition of a document visualisation stored at the link level. Each link to potentially the same document may have its own specific abstraction that is not necessarily derived from the destination document.
- Document defined abstracts are visualisations of a document that are defined in the Microcosm Document Management System. Any time there is a request to abstract a document without an overriding link abstraction this document abstraction is used. In the case of laserdisc and digital video micons can be used to abstract documents and thereby improve the user's ability to choose a destination video document. In the case of laserdisc video this is very useful as the video segments may be on a number of disks and visualisation removes the need to follow each link in order to determine the relevance of its content. Wherever large datasets exists, usually in terms of still images and video, this form of predefined abstraction reduces the navigational time taken to locate documents of interest. Where networks are in use, abstracts on local or remote machine reduce network traffic significantly during browsing.

- Finally if there is no abstract defined in the link specification or in the document management system, the OSH tries to generate an abstract dynamically. While this does not reduce any system fetch overhead, it allow the user to see simplified representations of documents in the context of each other and in many cases results in more effective navigation.

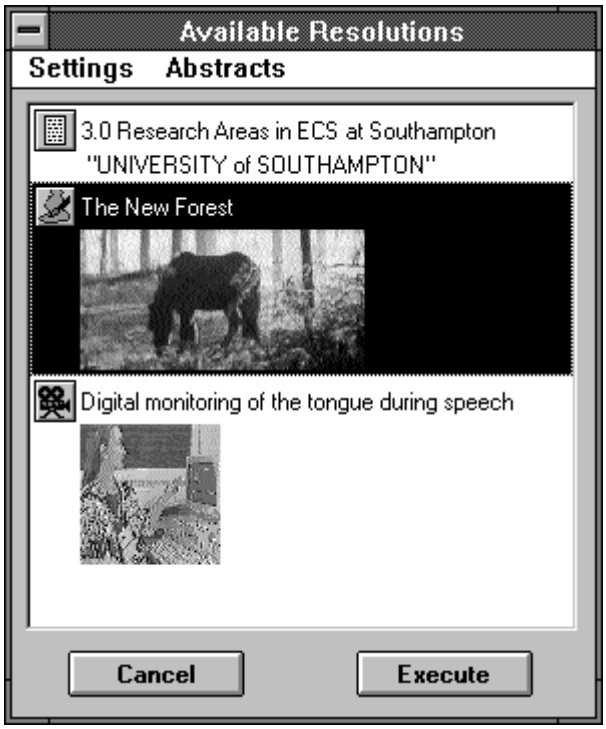

Fig 4. Available resolutions with document abstraction

Figure 4 shows abstraction for three documents. The first is plain text, has no layout and hence the destination selection is shown as the abstraction. The second document is a still image; the resulting abstract is derived at run time from the original document. The final abstract is for a piece of digital video, and a predefined Micon is used. Limited examples of document thumbnails can be seen in some of the more advanced multimedia database browsing systems[22, 8].

The scope for document abstraction is endless, and issues regarding text and audio are currently being considered. In cases where text documents have layout, a thumbnail of the destination page, or the document heading, may provide useful navigational clues. In sound documents various techniques can be used depending on the type of sound involved. A visual music score may be used to represent instrumental music, whereas a wave form could be used to represent other arbitrary noise. Then there is also the case of non visual representations for the documents being abstracted e.g. Gaver[10], Hardman[3].

The various features available, and the media that they are available for can easily be extended or modified by alteration to the OSH. Further development of abstraction techniques does not require changes to the available resolutions application. Likewise the inclusion of generic specification techniques does not affect the implementation of the link database, and matching algorithms are implemented at the OSH level.

Additionally, for some media not all services are currently supported, and the OSH utilises the most relevant technique available to service any object request. For a text type object such as RTF, a plain text representation is also included to allow services that only understant ASCII text to work with RTF objects.

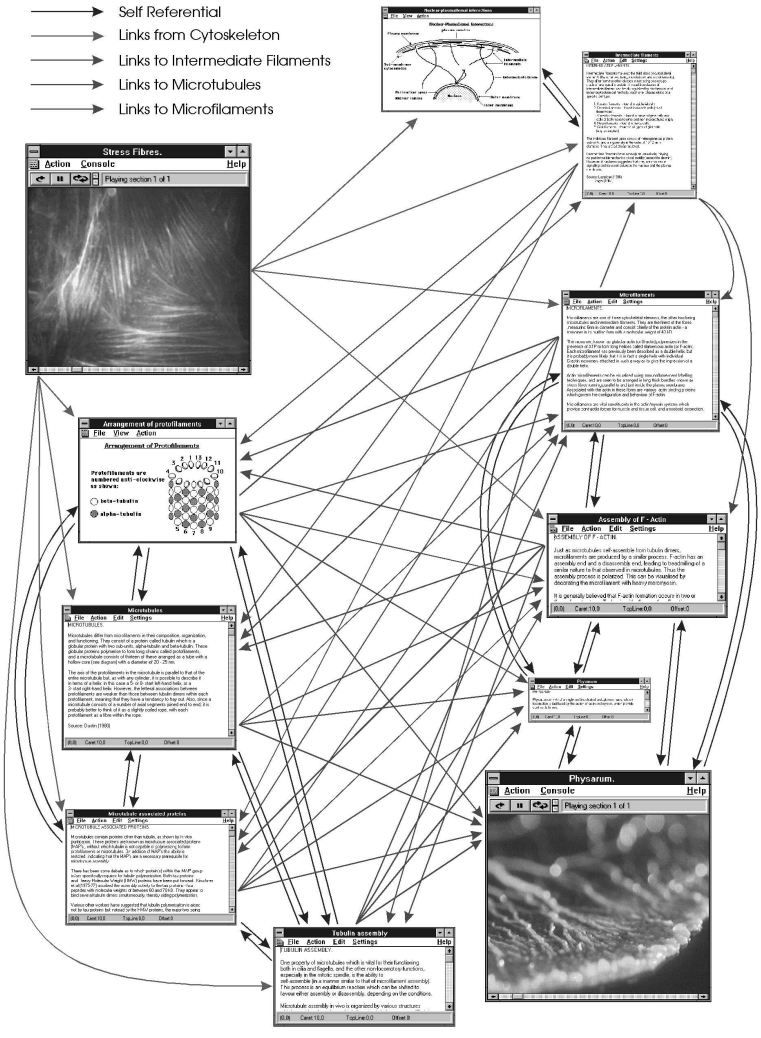

Fig 5. Explicit authoring of a small hypermedia network.

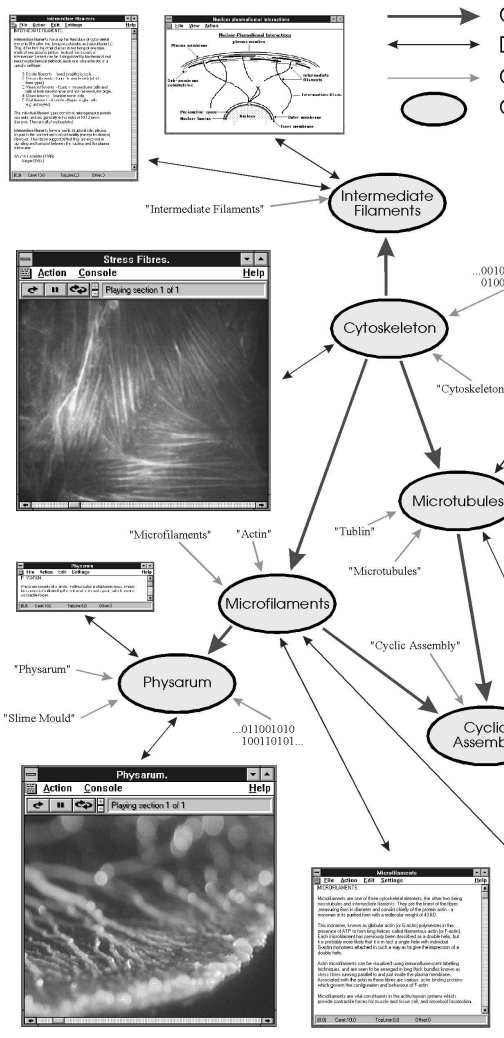

Fig 6. Implicit authoring using con

5 LINK ABSTRACTION

The development of the OSH was not only motivated by the desire to improve media consistency and support in hypermedia, but to improve the authoring techniques available.

Earlier in this document there was brief discussion about support for implicit authoring strategies across media. Links are physical rather than conceptual; linking from instantiations of an object in both text and image anchors to a destination requires two separate and independent links. There is currently no direct correlation between an implicit specification for an anchor in one medium and an anchor in another.

Systems like Hyperset[24] and Strathtutor[19], provide an interesting approach to link authoring which holds some useful strategies that may provide the solution to consistent authoring across media. Both these systems link at a document to document level, and use keywords or attributes to imply binding between documents. Working at the document level allows a degree of simplification that bypasses the media issues. Mechanisms like keyword association are used to cluster documents together. From any single document you can move to a set of *similar* documents that share common attributes.

Microcosm++ gives us an extensible platform to take these initail ideas and with the OSH extend them to provide conceptual content oriented links which are specified independently of any media. Whereas these systems are document based links, those in Microcosm++ are specified at any level of granularity that the appropriate media can support.

The OSH can be used to move from a media dependent representation space into the information domain where it then becomes possible to apply new types of link architecture.

Link abstraction has been moved to the higher level of *concept* to *concept* authoring. This opens up a completely new area of hypermedia, allowing integration of previously separate research topics e.g. semantic authoring[20], automatic document clustering techniques[25, 17, 16], and knowledge based systems, in a modular and unobtrusive manner.

Figures 5 & 6 show the structural difference, and reduction in authoring effort, between using explicit links and implicit links created at a conceptual level to describe relationships between documents. The structure in Figure 6 has at least the functionality of Figure 5 but is also future proofed and contains many more links than can be derived from the explicit definition. Conceptual links in Microcosm++ are still created manually, but are an expression in terms of the authors perspective and are independant of the physical content and data representation in the information space.

CONCLUSION

In this paper we have shown that the object based service architecture of Microcosm++ provides a significant improvement on the functionality offered by Microcosm. The current implementation of Microcosm++ demonstrates the flexibility of object based services and suggests a way forward for many new strategies to make hypermedia more viable in the workplace. The approach described here reduces significantly the authoring effort while at the same time increasing the integrity of any link structures.

The scope for link abstraction made possible by the Object Services Hierarchy leads to a consistent and uniform management of inter-media relationships. This promotes links to a status independent of the media and the documents involved, allowing minimal authoring to describe maximum connectivity of data.

ACKNOWLEDGEMENTS

Thanks to Gerard Hutchings for reviewing drafts of this paper, and everyone else in the Image & Multimedia Lab for their assistance. Some of this work has been sponsored by a grant from Marks & Spencers PLC who also provided material for the example application show in figues 2 and 3.

REFERENCES

1 Adobe Systems Incorporated, "Acrobat", Portable Document Format Reference Manual, Addison Wesley, June 1993.

2 Beitner, N.D. "Media support in Hypermedia", *Mini-Thesis*, Department of Electronics and Computer Science, University of Southampton, 12-29, July 1993.

3 Brewster, S.A., Write, P.C. & Edwards, A.D.N. "An evaluation of earcons for use in auditory human-computer interfaces", *In Human Factors in Computing Systems - INTERCHI'93 Conference Proceedings*, 222-227, April 1993.

4 Brøndmo, H.P. "The elastic tools: a descriptive overview", *Media Lab Working Paper*, MIT Media Laboratory, Cambridge Massachusetts, 1989.

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

6 Buchanan, M.C. & Zellweger, P.T. "Specifying Temporal Behaviour 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, November 1992.

7 Davis, H.C., Hall, W., Heath, I., Hill, G. & Wilkins, R. "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, November 1992

8 Egido, C. & Patterson, J. "Pictures and category labels as navigational aids for catalogue browsing", *In Proceedings of the ACM - CHI'88 (Human Factors in Computing Systems)*, 127-132, May 1988.

9 Fountain, A.M., Hall, W., Heath, I. & Davis, H.C. "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, 298-311, November 1990

10 Gaver, W.W. "Synthesising auditory icons", *In: Human Factors in Computing Systems - INTERCHI'93 Conference Proceedings*, 228-235, April 1993.

11 Goose, S. "Enriching Audio Support in a Hypermedia System", *Technical Report CSTR 94-03*, Department of Electronics and Computer Science, University of Southampton, 1994.

12 Halasz, F. & Schwartz, M. "The Dexter hypertext reference model", *In: Proceedings of the NIST Hypertext Standardisation Workshop,* Washington: US Government Printing Office, 95-133, January 1993.

13 Hardman, L., Bulterman, D.C.A. & van Rossum, G. "The Amsterdam Hypermedia Model: Extending Hypertext to Support Real Multimedia", *In: Hypermedia*, 5(1) 47-69, 1993.

14 Hardman, L., Bulterman, D.C.A. & van Rossum, G. "Links in Hypermedia: the Requirement for Context", *In: Proceedings of the fifth ACM Conference on Hypertext,* Seattle, November 1993.

15 Hill, G, Wilkins, R, & Hall, W. "Open and reconfigurable hypermedia systems: A filter based approach", *Technical Report CSTR 92-12, Department of Electronics and Computer Science, University of Southampton*, 1992

16 Hirata, K., Hara, Y., Shibata, N. & Hirabayashi, F. "Media-based Navigation for Hypermedia Systems, *In: Proceedings of the fifth ACM Conference on Hypertext,* Seattle, 82-93, November 1993.

17 Lee, S., Yang, M. & Chen, J. "Signature File as a Spatial Filter for Iconic Image Database", *In: Journal of Visual Languages and Computing,* Volume 3, 373-397, December 1992.

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

19 Mayes, T.J., Kibby, M.R. & Watson, H. "Strathtutor: The development and evaluation of a learning by browsing system on the Macintosh", *Computers in Education*, 12(1):221-229, 1988.

20 Nanard, J. & Nanard, M. "Should Anchors be Typed Too? - An Experiment with MacWeb", *In: Proceedings of the fifth ACM Conference on Hypertext,* Seattle, 51-62, November 1993.

21 Newcomb, S.E., Kipp, N.A. & Newcomb, V.T., "The HyTime hypermedia/time-based document structuring language. *In: Communications of the ACM,* 34(11):67-83, November 1991.

22 Niblack, W., Barber R., Equitz, W., Flickner, M., Glasman, E., Petkovic, D., Yanker, P. & Faloutsos, C. "The QBIC Project: Querying images by content using colour, texture, and shape", *RJ 9203 (81511) IBM Research Report,* February 1 1993.

23 Palaniappan, M., Yankelovich, N., Sawtelle, M. "Linking active anchors: a stage in the evolution of hypermedia", *In: Hypermedia,* 2(1):47-66, 1990.

24 Parunak, H.V.D. (1991). Don't Link Me In: Set Based Hypermedia for Taxonomic Reasoning. *Hypertext'91 Proceedings*. ACM Press.

25 Salton, G., Allan, J., "Selective Text Utilisation and Text Travesial", *In: Proceedings of the fifth ACM Conference on Hypertext,* Seattle, 131-144, November 1993.

26 Wilkins, R.J., Griffiths, S.R., Hall, W., Lewis, P.H., Davis, H.C., “The MAVIS Project - Extending generic links and content based retrieval to no-textual documents in the Microcosm model.”, *1994 Research Journal, Department of Electronics and Computer Science, University of Southampton.*

27 Yankelovich, N., Haan, B.J., Meyrowitz, N.K. & Drucker, M. (1988) Intermedia: The Concepts and the Construction of a Seamless Environment*. IEEE Computer,* January 1988.