Extending the Microcosm Model to a Distributed Environment

Gary Hill, Wendy Hall
University of Southampton,
Highfield, Hants, UK
SO9 5NH
fax : +44 703 592865
e-mail: {gjh,wh}@ecs.soton.ac.uk

ABSTRACT

In recent years, there has been significant growth in the use of computer networks to support electronic delivery of information. As the volume of available information has grown, a need for powerful tools that can manage access has arisen. It has been suggested that hypertext techniques can provide such a facility.

The Microcosm system is a hypertext link service developed at the University of Southampton. The system is based upon a modular architecture which allows the functionality of the system to be easily and dynamically extended. This paper describes the development of a distributed version of Microcosm based upon this modular design.

The distributed system described utilises the fine granularity of the Microcosm model to support a wide range of possible configurations. The system also extends the document management facilities of Microcosm to allow information stored by other information services to be incorporated. The result is a system that can apply Microcosm's open linking services to a wide range of networked information.

KEYWORDS: Open, Distributed, Hypertext, Microcosm.

Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the ACM copyright notice and the title of the publication and its date appear, and notice is given that copyright is by permission of the Association for Computing Machinery. To copy otherwise, or to republish, requires a fee and/or specific permission.

© 1994 ACM 0-89791-xxx-x/xx/xxxx...

1. INTRODUCTION

As the field of hypertext and hypermedia has grown in recent years, there has been increasing interest in the creation of 'open' hypertext systems and the advantages of such an approach have been widely addressed, in particular by Malcolm et al in their paper at Hypertext '91 [9]. There is no precise definition of the features of which such systems are comprised, but there are many aspects which are generally accepted to be desirable attributes. The definition which we have used in our own research in this area was presented at ECHT '92 [3], and is based on the following requirements:

1. A system which does not impose mark-up on the data being linked

2. A system which is able to integrate tools that already exist within the host environment, and which can utilise data created with such tools without compromising the continued use of the data by the creating tool.

3. A system which allows data and processes to be distributed across a network, and across heterogeneous hardware platforms.

4. A system with no artificial distinction between author and reader.

5. A system which will allow new functionality to be easily incorporated.

These requirements have been used as a basis for the design of the Microcosm system which has been under development at the University of Southampton for around 4 years. The version of Microcosm currently in use, described in section 3, meets all of these requirements except for 3. Although the system was designed with distributed operation in mind, the current implementation provides a single-user servicfe, and can only share data

through the use of simple, LAN-based file sharing facilities.

As access to computer networks becomes more commonplace, the ability of an information system to operate in a distributed environment, and to incorporate other forms of distributed system, is likely to become increasingly important. The non-intrusive linking facilities of an truly open, distributed, hypertext system would offer an ideal way of providing flexible access to an ever-increasing variety of on-line information.

This paper briefly reviews some examples of distributed hypertext systems, and then describes the extension of the Microcosm link service to allow it to operate in a distributed environment. The resultant model is designed to compliment the flexibility of the Microcosm filter model, by offering a range of possible network configurations to suit a variety of applications.

2. EXISTING DISTRIBUTED HYPERTEXT SYSTEMS

There are already numerous hypertext systems which can operate in a distributed environment. This section describes some notable examples and compares their features to the requirements outlined in the previous section.

As networks such as the Internet have become more accessible in recent years (figure 1), there has been a rapid growth in interest in easily accessible, distributed information services. The outcome of this has been the development of information server systems such as WAIS [8], Gopher [10] and the World-Wide Web (WWW)[1].

A weakness of such systems is that they tend to offer very structured and inflexible views of information, but the success of the WWW has shown that these large information resources can be made much more accessible and effective through the use of hypertext links.

WWW is probably the most well known and widely used distributed hypertext system currently available today. Conceived at the European Particle Physics Laboratory, CERN, in Switzerland, and popularised by the US National Centre for Supercomputing Applications (NCSA) with the development of the Mosaic client application, it provides access to a wide range of information. A WWW document is based upon the Hypertext Markup Language (HTML) which is a simple variation of SGML, oriented towards the description of hypertext links.

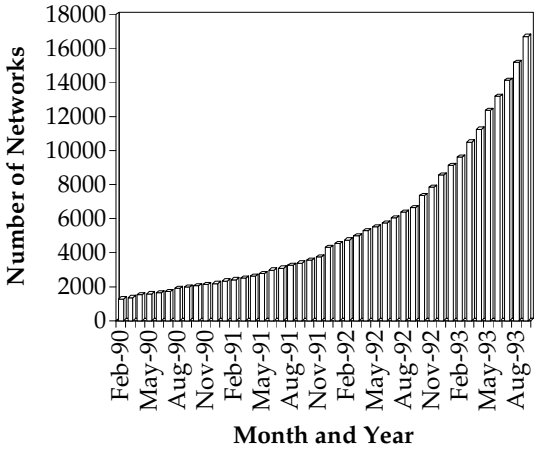

Figure 1: The growth of the Internet by network between February 1990 and August 1993 . The graph is compiled from statistics maintained by NSF. Statistics about the Internet and its traffic are available by anonymous FTP from nic.merit.edu as /nsfnet/statistics/history.netcount

The ability of WWW servers and clients to interoperate between many heterogeneous hardware and software environments, and to incorporate information from a variety of different sources, mean that it can be classified as an open system. However, it is clear from the definition given in the previous section that WWW servers as currently implemented cannot be considered to provide an open *hypertext* system. Although it is distributed, it is not possible to dynamically adjust the available functionality, links are encoded as markup within HTML documents, and there is a clear distinction between the reading and authoring of WWW documents. This makes it impractical as the basis of an open hypertext service in its current form.

The Virtual Notebook System [13, 2], or VNS, is a hypertext system designed to allow information acquisition and sharing, in particular for scientific communities. As such, distributed access to the information available is an important aspect of its functionality. VNS is based on the relational database SyBase as a hypertext engine, with specially developed presentation components running under X-Windows.

The distributed model of the VNS is based on work groups, which each have a work group server (WGS). The WGS can also act as a gateway to other VNS work groups, or to external information services. Additionally the system is further enhanced by the ability to incorporate other information systems via 'gateways' in the server.

Sun's Link Service [12], is a well known and groundbreaking example of an open hypertext system. Running on Sun workstations (although the open protocol suggests that a heterogeneous implementation would be possible), in a distributed workstation environment, it consists of a link database service that performs all link management tasks centrally. Applications use a simple protocol to communicate with the Link Service. A notable feature of the Link Service is that all linking is managed separately from documents, without the use of mark-up.

Although tools may easily be extended to incorporate additional viewers, there are no facilities to extend the functionality of the Link Service to offer other forms of linking.

Hyperform [14] is a distributed hypertext system designed to provide an extensible range of hypertext services. Rather than provide a fixed, general purpose system model, the Hyperform server provides a set of object classes that can be used to tailor the hypertext functionality available. Thus Hyperform is one of the few systems that meet the fifth requirement described in section 1, to be able to easily extend the underlying hypertext service. Similarly, the presentation layer of the system is configurable through the use of integrating processes which allow external applications to be incorporated into the system.

The systems described above illustrate just some of the potential distributed models that are possible. Sun's Link Service and Hyperform are able to offer a hypertext service to a wide range of applications, but do not address the issue of wider distribution. They are based on the use of a central server accessed by all users. The WWW is very different, being designed around the concept of widely distributed servers which may all be interlinked via the documents which they manage. However, its information base tends to be static in nature due to the way that links must be embedded. The VNS system can be viewed as a hybrid of these two approaches, with an emphasis on small work groups which can share information via a local server, but also providing the facilities for information to be shared between work groups.

Similarly, the type of use to which these different systems are put varies a great deal. The WWW, due to its distinction between authoring for, and using the system, is oriented towards browsing. At the opposite extreme is the Hyperform system which provides a wide range of facilities for the support of co-operative work and authoring. Sun's Link Service offers simple hypertext facilities to the user, but does not have the in-depth support for cooperation provided by Hyperform. Again, VNS offers a compromise between these extremes, with support for co-operation, but also for straightforward access to information sources (e.g. WAIS queries).

The varying facilities and usage of these systems illustrate some of the ways in which an open, distributed hypertext system might be required to function. Clearly, in conjunction with the flexible framework required to provide modular hypertext functionality and integration of applications, a flexible approach to the distribution of functionality and data is required. The following sections briefly summarise the features of Microcosm, and then describe the design of a distributed model which aims to complement the flexible nature of the Microcosm model.

3. THE MICROCOSM SYSTEM

Microcosm is an open hypertext system designed to meet the requirements described in section 1. It comprises a modular, extensible architecture that allows both the underlying hypertext functionality, and the presentation components of the system, to be tailored to suit a user's requirements. The logical model of this architecture is shown in figure 2.

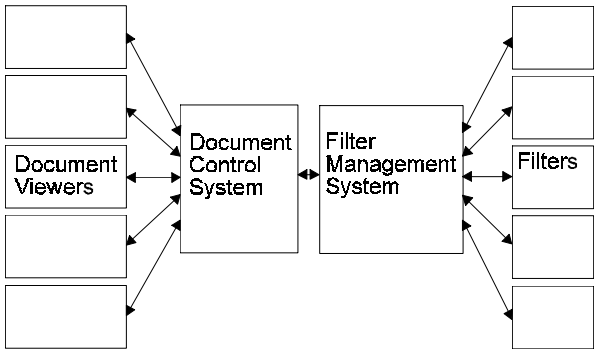

Figure 2: Microcosm 2.0 System Model

The Document Control System (DCS) manages the document viewer processes that provide presentation of the available information. Viewer processes may be written specifically for Microcosm, or third party applications may be integrated dependent upon their flexibility [3]. This allows Microcosm to provide an underlying hypertext service to a user's preferred applications. The facilities provided by Microcosm are accessed by sending messages to viewers which request certain actions to be carried out (e.g. Follow Link).

The remaining components of the system provide the actual hypertext layer. As users interact with the information presented in viewers, messages are sent to the DCS. This in turn passes messages to the Filter Management System (FMS). The FMS provides hypertext

functionality through the use of a number of modules known as filters.

As the FMS receives messages from the DCS, it passes them to the set of active filters. When a filter receives a message, it examines the type of the message and, if appropriate, takes further action. For example, a filter which implements a link database will respond to a Follow Link message with a series of messages describing any applicable links. Alternatively, a filter storing a history of a user's actions would simply record details of any Open Document messages received.

The original design of the FMS arranged the active filters into a linear chain. Messages are sent to the first filter in the chain, and then, as messages are received from filters, they are passed on to the next filter in the chain. As messages reach the end of the chain, the FMS passes them back to the DCS which will pass them onto a viewer if necessary. A more detailed description of the provision of hypertext functionality via this filter model is given in [6].

Microcosm was initially implemented as a purely stand-alone system, and had little scope for use in a distributed environment. Such use was limited to the distribution of information via local area networks, and the use of network aware applications as viewers or filters. However, the system was designed for eventual use in a distributed environment, and the modular architecture of the system is obviously well suited to the creation of a truly distributed version of the system.

The chain-based implementation of the FMS, however, is not ideal for such an extension of its role. The use of a linear chain of filters requires all messages to be routed through all active filters, even though not all filters understand all message types. This is obviously inefficient, although the stand-alone system is still able to perform satisfactorily. However, it would cause a significant reduction in the performance of a distributed system, since messages would take longer to deliver across a network, and the inefficiencies of the system would be more apparent.

In order to address this limitation, a more advanced version of the FMS was devised. This system requires the active filters to register the type of messages which they will accept, and thus allows the FMS to build a table of the actions available and the filters which process them. In effect, the FMS is now able to maintain several small chains, one for each available action, instead of one compound chain. This approach also has other benefits, such as reducing the need to take care of filter ordering to ensure that filters receive the messages they need.

The resulting message-routing table may then be used by the FMS to deliver messages of a particular type intelligently, sending them only to those filters which are able to process them. Within the chain for each action, the message is treated in the same way as with the initial, single chain, system. However, if the filters generate messages of different types, these are transferred to the appropriate action chain, if there are filters which process that type, or back to the DCS. A full description of the limitations of the simple, chain-based FMS and the development of the more advanced system is given in [7].

With this more efficient message routing system, the creation of a filter-based distributed system is much more feasible. The following section describes the design of the system that has been developed and discusses the range of distributed models that it makes possible.

4. DISTRIBUTION OF OPEN HYPERTEXT FUNCTIONALITY

4.1. Requirements of a Distributed Hypertext System

The previous section has shown briefly how the modular filter architecture which provides the basis of Microcosm's hypertext functionality allows the system to be easily, and dynamically, configured to support particular user requirements. The aim of the distributed version of Microcosm is to provide this configurability in a networked environment, and to allow a complementary level of flexibility in the distributed model created.

The definition of an open hypertext system given in section 1 includes a requirement that systems should offer distributed functionality in a heterogeneous environment. This requirement clearly allows a range of interpretations as to the meaning of distributing hypertext functionality and information. This suggests that a flexible approach is required, and obviously, for heterogeneous platforms to be integrated, an open protocol based upon a widely available network communication system is a basic requirement.

Another requirement of the definition is for systems to be able to utilise existing tools within the host environment. As well as being concerned with basic tools such as word processors and spreadsheets, this requirement also applies to the incorporation of other distributed systems. A user of a system who wishes to utilise the facilities of distributed information services such as WAIS and Gopher should be able to use their particular hypertext system in conjunction with these facilities; for example to make links to and from information accessed with these tools.

The flexible, open model provided by Microcosm is well suited to meeting both of these requirements:

- Existing information services may easily be incorporated as either filters or viewers depending on the type of functionality provided. For example, the browsing-oriented interface of Gopher could be incorporated as a viewer. As such, it might offer 'meta-viewer' functionality, launching other Microcosm viewers to display the information located. A query-based system such as WAIS, on the other hand, is more suited to incorporation as a filter; this would allow a powerful searching facility to be provided, to compliment authored links.

- The distribution of Microcosm's own functionality obviously requires extensions to the existing system, however the modular nature of the system architecture is an ideal basis for the provision of such extensions. The modular way in which hypertext functionality is provided permits distribution to take place at the filter level, allowing discrete subsets of the total functionality to be made available. Similarly, access to remote documents can easily be incorporated, since all such access is made via Microcosm's document database system, know as the Document Management System, or DMS. This system can be modified to identify those calls which refer to remote documents, and take appropriate action.

As outlined in the previous section, the filter model used by Microcosm allows a great deal of flexibility in the type of hypertext functionality that is offered. By basing the distributed version of the system upon this flexible approach, it is able to provide an equally flexible approach to the provision of shared functionality. The discussion of existing distributed hypertext systems in section 2 illustrated the various approaches to this problem that have been utilised in the past. However, the distributed functionality of the systems described was restricted due to the fixed distribution of components permitted by the system models utilised. In particular, each system tended to be based upon a simple single server model, with hypertext functionality provided as a central service, and users utilising client applications on their personal workstations.

Whilst there is clearly a need for the provision of centralised functionality, as with typical client-server configurations, a more flexible system would also allow more direct co-operation between host systems and users.

4.2. The Distributed Microcosm Model

The distributed version of Microcosm aims to provide a flexible range of configurations based upon the filter process as a unit of distribution. This allows the flexibility of configuration provided by the filter model to be inherited by the distributed version of the system; for example, allowing a particular subset of the local filter configuration to be made available to remote systems, or conversely, assimilating remote filters into the local filter set. Since the FMS manages the routing of messages to filters, it is the logical place to locate the extensions required to handle a system where filters may be accessed from remote systems.

To provide such functionality, the FMS must be extended so that it allows local filters to be 'published', allowing other systems to utilise them, and so that remote published filters can be connected to the local system. The FMS can then utilise a well-defined protocol to transfer messages to remote systems when necessary. Similarly, each FMS must be prepared to receive messages from remote systems, either in reply to a message sent to a filter within that system, or a message to be delivered to a published filter in the local filter configuration.

A system such as this allows links, and other forms of hypertext functionality, to be accessed from a remote system. Such activity is likely to lead to references to documents stored on other systems. Therefore, a complementary extension to the system is required to allow such documents to be retrieved. The Microcosm DMS is the approved approach for accessing the file system, this is therefore the logical place for distributed document access to be incorporated. In order to determine whether a document identifier refers to a local or remote file, the document identifier is simply extended to incorporate details of the host system. This allows attempted access to remote files to be easily detected, and facilitated using a standard protocol.

4.2.1. Distribution of Hypertext Functionality using Filters

As described above, the basic element of distribution for Microcosm is the filter. In the simplest case, two users may collaborate through the use of shared Linkbases, each providing the other with links to annotations and personal documents. However, since filter processes represent a small system granularity, a wide range of distributed models is possible through the use of this system. These configurations are summarised in table 1.

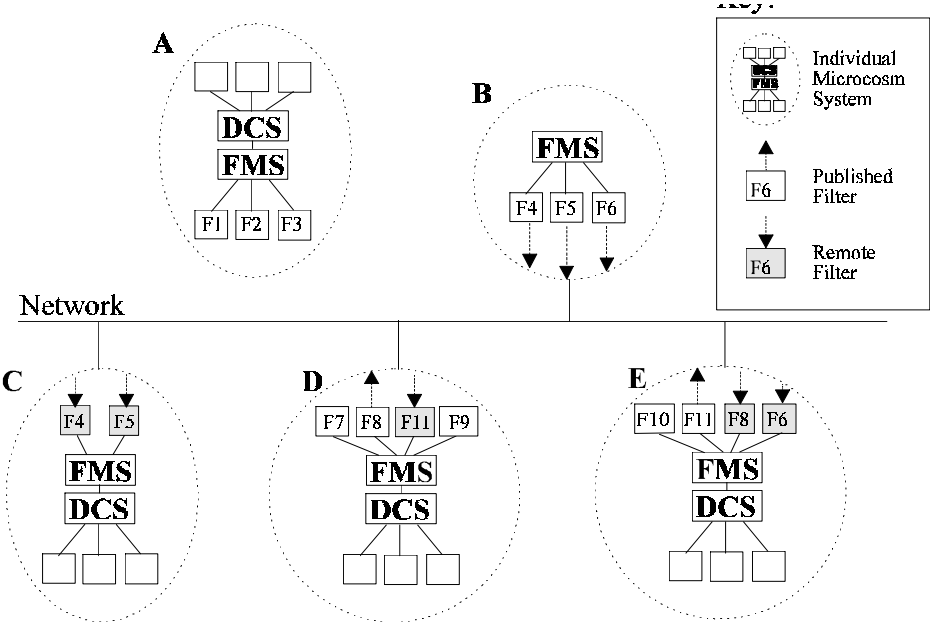

Figure 3: The range of possible filter configurations available using the Microcosm distributed model.

Distributed Model for local system	How the system operates
No Distribution	The system operates as described in earlier chapters, with no interaction with remote systems
Client Only	The system operates purely as a client, with all hypermedia functionality provided by a remote server(s)
Server Only	Alternatively the system might make all of its functionality available to remote systems and disable local interaction with the system, operating purely as a server.
Peer to peer interaction	Finally, the system can combine both client and server functionality within the same configuration. This creates a network comprised of equal, peerless systems.

Table 1: The range of distributed architectures available by utilising the flexibility of the Microcosm distributed model.

The table shows how, by varying the approach taken to publishing and connecting distributed filters, a whole spectrum of models is possible. Systems based upon distinct client and server components, such as WWW or Sun's Link Service, can easily be simulated by configuring particular Microcosm systems as either server-only or client-only. Similarly, it is still possible for Microcosm to act as a stand-alone system. There is no system-enforced requirement for the distributed facilities to be used, they simply offer an enhancement to the basic functionality of Microcosm.

In between these two extreme examples (pure client/server interaction and stand-alone system), lie a whole spectrum of possible configurations. These are based upon the peerless nature of the model that is achieved, where any particular Microcosm system on a network may act as an information server and a client system within the same configuration.

This brings much more flexibility than the strict client/server transactions provided by other distributed hypertext systems, offering scope for many forms of collaboration between users or work groups. Additionally, since all these configurations are possible within the same system, and each is based on a well-defined, open protocol, all configuration variations may co-exist. Thus any particular system can be configured in such a way to suit its particular purpose, whilst not excluding its use by other systems which are used in alternative configurations. The diagram in figure 3 illustrates the wide range of configurations that have been described, and shows how the various possible configurations are able to interact.

System A shows Microcosm operating in a stand-alone manner as was possible in previous versions of the system. System B shows Microcosm acting as a simple hypertext server with no local interaction. System C shows Microcosm acting purely as a client, obtaining all hypertext functionality from remote filters (in this case provided by system B). Finally, systems D and E show two Microcosm systems operating in a peer to peer manner, they are sharing filters to allow co-operation between the systems in addition to incorporating local filters and remote filters from other Microcosm systems.

4.2.2. Access to Remote Documents

As described earlier, access to remote documents can be provided by extending the Microcosm DMS. The main extension required is the ability to distinguish document identifiers which refer to remote systems, from those which specify documents available on the local system. The current unique document identifier is constructed using the date and time that a document is imported into the database. In order to identify the network location of a

document, it is necessary to identify the host system which is storing the document in addition to the document identifier.

The format chosen was the Universal Resource Locator (URL) used by the World Wide Web system. This is a general purpose structure that allows a variety of access protocols to be specified. The document identifier in the distributed Microcosm system is formed as follows:

<access method>:\\<network name>[:<port number>]\<locally unique id>

For Microcosm-managed documents, the access method is always set to "mcm", but by incorporating this field other access protocols may also be supported by Microcosm. This allows links to be made to documents stored in other systems, for example WWW (access mode "http) or FTP (access mode "ftp"). Typically, the local unique id for a Microcosm document will be the DMS identifier, but it can also be a simple file name. This is the key difference from the standard WWW URL definition, although this portion of the URL may be constructed in many ways, depending on the server, it is typically just used to specify a filename. The advantage of the use of an abstract document identifier is that the actual file may be moved or renamed on the local system, but still be accessible from remote systems, since the DMS on the host system will record any change of location. With a standard URL, hard-coded document references can easily become out of date if files are moved, a factor which can affect the integrity of WWW documents.

To utilise this extended document identifier, the database function that creates a new identifier must be updated, and any functions which use such an identifier must extract the access method and host details to check whether the document being referenced is present on the local system. If not, a standard protocol is used between databases on distinct systems to transfer the appropriate file, or its database record.

To accommodate the use of a variety of access protocols within the distributed system, the DMS has been extended in a modular manner. Separate modules can be provided to implement a particular type of access, and loaded dynamically as required. This reduces the amount of effort required to incorporate new access protocols into the system.

5. IMPLEMENTATION

5.1. Network Protocol

One of the key requirements of the distributed version of Microcosm described in this paper is that it should be able to operate in a network environment comprised of heterogeneous hardware and software platforms; a process that is often fraught with incompatibilities and inconsistencies. In order to do this, the communication and connectivity techniques used must be as widely available as possible. For this reason the development has been based on standard network protocols. However, the actual design is not reliant on such protocols being available, and the network specific modules of the system could easily be altered to support other platforms. The only requirement is for some form of general purpose, connection-oriented network transport layer to be available.

The only aspect of the underlying network to directly affect the implementation of the distributed version of Microcosm is the actual communication protocol used between instances of the system. TCP/IP protocols are used because of their widespread availability on many platforms (the protocol is a common element of UNIX-based systems, and is available for both IBM compatible and Macintosh personal computers), and the fact that it is already used by a large number of existing Internet information services (e.g. WAIS, Gopher, WWW, Archie). Although other protocols may offer solutions for particular platforms (NetBIOS for IBM, Appletalk for Macintosh), no other protocol is available for such a wide range of systems.

Although the only current implementation of the distributed Microcosm system is for the Windows environment, the use of TCP/IP as the basis for the networking functions means that the creation of versions for alternative platforms such as Apple Macintosh and X-Windows is simply a matter of adhering to the appropriate protocols. Versions of Microcosm for these platforms have been developed, and the incorporation of the distributed services described in this chapter will be investigated once they reach a more mature state.

An additional advantage of open protocols that they are not restricted to a closed set of conforming applications. If the details of the protocol are well known, other applications will be able to utilise the functionality of Microcosm, without needing to provide the full Microcosm framework.

5.2. Inter-FMS Communication

In order to support the model described in section 4, the FMS has been extended to allow communication with other network hosts running the distributed version of Microcosm, using the facilities outlined in the previous section.

The first step is to locate remote Microcosm systems, and establish whether these systems have filters that may be incorporated in the local configuration. This requires the following queries to be accommodated by the distributed FMS:

- Does a particular host offer a Microcosm server?
- Does a particular server have any published filters?
- What are the details of a particular filter, so that a virtual connection to it may be created? For example, what type of messages will it accept, and the network port to be used.

These queries are delivered to a 'well known' network port, thus ensuring that if a particular system offers Microcosm services, then other Microcosm systems can locate them. Rather than maintain constant connections to remote filters (which would burden systems in wide use), the necessary details to contact remote systems when necessary are stored.

At present the systems to which theses queries are delivered to are determined either by sending to a list of known hosts, or by broadcasting to all local systems. A more advanced system could incorporate a distributed naming service for Microcosm servers with a query system to locate appropriate servers.

Once remote filters have been integrated into a particular system's configuration, messages generated within the system will eventually need to be transferred to these remote systems. These messages are delivered to a port specified when the remote filter was 'connected'.

In order to allow each FMS to establish whether a message it is dealing with originated from a local or remote system, additional fields identifying the message source are added to messages before they are transferred. This allows the remote FMS to route the message only to the appropriate filter. If a filter generates additional messages, any such fields must be copied to them. This allows messages to be returned to the source system once processed.

5.3. Distributed Document Management System

Similar extensions have been made to the Microcosm DMS to allow the retrieval of documents from remote systems. Since documents can be quite large, and therefore take some time to retrieve, each system maintains a short-term cache of documents to increase the performance of the system.

5.4. User Feedback

Finally, since the system will appear slightly slower when operating across a network, a 'networking now' signal is provided using a special cursor, to avoid users becoming concerned with the performance of their system.

6. CONCLUSIONS

The Microcosm system is an open, hypermedia system designed in a modular manner in order to allow the straightforward incorporation of new functionality. This paper has described the extension of this model to create of a distributed hypertext system which can exploit this modularity. The resultant system is able to provide a flexible range of distributed configurations, allowing the system to be utilised in a wide variety of ways.

The model described in this paper has been implemented for the Microsoft Windows version of Microcosm. Local experimentation with this implementation has successfully demonstrated that the model is able to support the flexible range of distributed models described above, and also integrate with a variety of other information services. The peer-to-peer nature of the system supports many different configurations, for example allowing users to collaborate with each other by sharing link databases and documents, or providing a central hypertext storage service which users may incorporate into their local configurations.

This flexibility allows users much greater choice in the way their hypertext environment is configured and utilised, rather than providing a single distributed model with fixed functionality and constraints within which users must operate. Experiments with a wide area network have been less extensive, however, results obtained show little change other than a performance penalty.

In the same way that the stand-alone version of Microcosm has been used as a test- bed for the development of open hypertext functionality, this initial implementation of the distributed version offers an ideal basis for further investigation into the provision of distributed hypertext services. For example, additional facilities such as notification control and locking [4, 15] within the document database would add greater support for the co-operative development of documents, and the addition of security facilities would enable filters to be made available to selected groups of users only.

A common weakness with the current generation of hypermedia systems is in support for versioning of information [11, 5], and this is particularly important in a distributed system, where multiple, simultaneous users are able to access the same information, and the offsets of link

sources and destinations are liable to change. This problem is equally applicable to Microcosm, although the distributed system is still effective without versioning support.

Other developments in network technology will also affect the future development of the system. Advances in hardware such as ATM networking offer much greater potential for distribution of true multimedia data (e.g. full motion digital video and high quality audio), while new support systems for distributed applications (e.g. DCE and CORBA) promise to provide many useful underlying facilities.

REFERENCES

[1] Berners-Lee, T., Cailliau, R., Groff, J., Pollermann, B., World-Wide Web: The Information Universe. CERN, 1211 Geneva 23, Switzerland, 1992.

[2] Burger, A. M., Meyer, B. D., Jung, C. P., Long, K. B., The Virtual Notebook System, In: Hypertext 91: Proceedings of Third ACM Conference on Hypertext (San Antonio, TX Dec. 15-18), 395 - 402.

[3] Davis, H., Hall, W., Heath, I., Hill, G. *and* Wilkins, R., Towards an Integrated Information Environment with Open Hypermedia Systems. *In:* D. Lucarella, J. Nanard, M. Nanard *and* P. Paolini, *eds ECHT '92.* Proceedings of the Fourth ACM Conference on Hypertext, Milan, Italy, November 30-December 4, 1992. ACM Press, 181-190.

[4] Grønbæk, K., Hem, J.A., Madsen, O.L. *and* SLOTH, L., Designing Dexter-based Cooperative Hypermedia Systems. *In:* Proceedings of the Fifth ACM Conference on Hypertext, Seattle, WA, November 14-18, 1993, ACM Press, 25-38.

[5] Haake, A., CoVer: A Contextual Version Server for Hypertext Applications, *In:* D. Lucarella, J. Nanard, M. Nanard *and* P. Paolini, *eds ECHT '92.* Proceedings of the Fourth ACM Conference on Hypertext (Milan, Italy, November 30-December 4, 1992), ACM Press, 43-52.

[6] Hill, G., Wilkins, R. *and* Hall, W., Open and Reconfigurable Hypermedia Systems: A Filter Based Model. Hypermedia, 5(2), 1993, 103-118

[7] Hill, G., Extending an Open Hypermedia System to a Distributed Environment, Ph.D. Thesis, Department of Electronics and Computer Science, University of Southampton, UK, 1994.

[8] Kahle, B., Wide Area Information Server Concepts, Thinking Machines, 1989.

[9] Malcolm, K.C., Poltrock, S.E., Schuler, D. Industrial Strength Hypermedia: Requirements for a Large Engineering Enterprise. *In:* Hypertext 91: Proceedings of Third ACM Conference on Hypertext, San Antonio, TX. ACM Press, 1991, 13 - 24.

[10] Obraczka, K., Danzig, P.B., Li, S., Internet Resource Discovery Systems, IEEE Computer, September, 1993, 26(9), 8-22.

[11] Østerbye, K., Structural and Cognitive Problems in Providing Version Control for Hypertext, *In:* D. Lucarella, J. Nanard, M. Nanard *and* P. Paolini, *eds ECHT '92.* Proceedings of the Fourth ACM Conference on Hypertext, Milan, Italy, November 30-December 4, 1992. ACM Press, 33-42.

[12] Pearl, A. Sun's Link Service: A Protocol for Open Linking. *In:* Hypertext '89 Proceedings, Pittsburgh PA, 1989, 137 - 146.

[13] Shipman F.M. III, Chaney R.J., Gorry G.A. Distributed Hypertext for Collaborative Research: The Virtual Notebook System. *In:* Hypertext '89 Proceedings, Pittsburgh PA, 1989, 129 - 136.

[14] Wiil, U.K., Leggett, J., Hyperform: Using Extensibility to Develop dynamic, Open and Distributed Hypertext Systems. *In:* D. Lucarella, J. Nanard, M. Nanard *and* P. Paolini, *eds ECHT '92.* Proceedings of the Fourth ACM Conference on Hypertext, Milan, Italy, November 30-December 4, 1992. ACM Press, 251-261.

[15] Wiil, U.K., Leggett, J., Concurrency Control in Collaborative Hypertext Systems, *In:* Proceedings of the Fifth ACM Conference on Hypertext, Seattle, WA, November 14-18, 1993, ACM Press, 14-24.