Technical Briefing

The Microcosm Link Service

Wendy Hall,
Gary Hill and
Hugh Davis
Department of. Electronics and
Computer Science,
University of Southampton,
Highfield, Hampshire,UK,
SO9 5NH.
wh@ecs.soton.ac.uk
gjh@ecs.soton.ac.uk
hcd@ecs.soton.ac.uk

KEYWORDS

Open, Integrated, Link Service

INTRODUCTION

As understanding of hypertext concepts has increased, the possibilities they present for the creation of a flexible system able to act as an integrator for a user's whole environment are growing. This has led to interest in the concept of open hypermedia systems that are able to act as a link service to an existing set of applications. The key to this concept is in the perception of hypertext principles as a method for integration rather than simply as a delivery medium for a clearly defined information set.

A vision of the possibilities such a system might provide in a real world situation is given by Malcolm et al. [Malc91]. They paint a convincing picture of an environment where hypertext principles provide a seamless integration of a diverse range of applications. In order to provide such facilities in an efficient manner, a common link service such as described by Pearl [Pear89] and Rizk [Rizk92] is essential. This would provide a standard linking protocol that any application can take advantage of in order to become a fully integrated part of its environment. The Microcosm link service which we have been developing at the University of Southampton addresses the need for an integrating technology of this type.

DESIGN OF MICROCOSM

THE MICROCOSM SYSTEM MODEL

Microcosm's design emphasises two main requirements, the ability to use existing applications whenever possible, through the separation of links from information and the provision of an extensible hypertext functionality through its modular underlying model.

The model consists of two distinct layers. Users see an interface that consists largely of their standard applications, co-ordination of these applications is provided by the Document Control System or DCS. The DCS acts as a mediator to the Filter Management System (FMS) that provides the actual hypertext functionality. As users' interactions with applications generate hypertext actions, messages are sent to the DCS and in turn to the FMS. This may in turn result in the activation of links, which leads to the FMS sending messages back to the DCS instructing it to open new documents.

Just as the DCS simply implements an open protocol for the co-ordination of document views, the FMS does not implement hypertext functions itself, but offers a similar protocol for the control of small processes or 'filters' that implement particular functions. This allows unrestricted growth of the functionality that can be provided by Microcosm. An example is the incorporation of navigation tools as described in [Hill93] or the addition of text retrieval algorithms to supplement traditional hypertext links described in [Li92].

The model described above is shown in figure 1, more detailed discussion of the model and its design can be found in [Hill93] and [Davi92] and will be presented in the briefing. The following sections discuss the range of issues involved in integrating a wide range of existing applications.

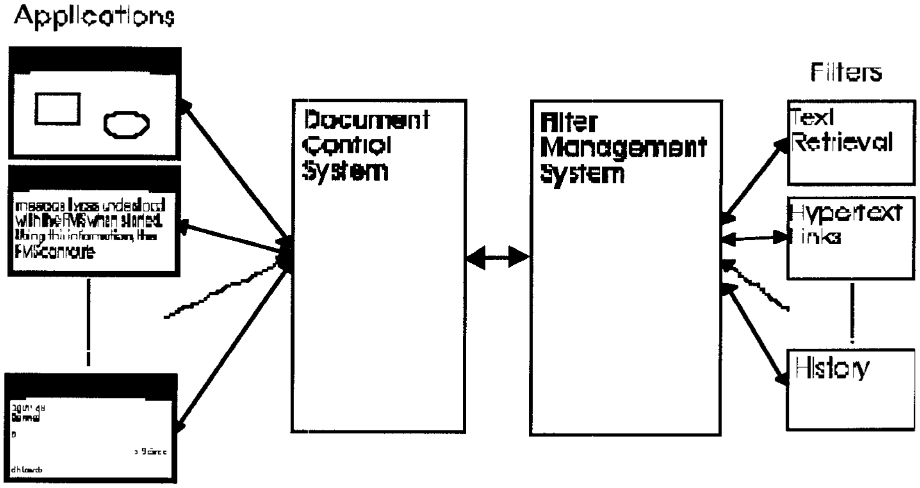

Figure 1: The Microcosm Model

LEVELS OF INTEGRATION

A *viewer* in Microcosm is any program which is able to display some format of data, and which can communicate with the Microcosm link service. This requirement to communicate with the link service has been a problem for systems that attempt to use third party applications as viewers, for example, Sun's Link Service [Pear89], required that all applications that wished to use the link service should be modified to become "link service aware". Pearl argues that if a link service was a standard feature of the operating environment, then all serious applications would be written to make use of this feature. However, in the immediate future this is not likely to be the case and so users of the system are limited to using those applications, such as TextEdit, which have been specifically modified.

For a viewer to be link service aware in Microcosm, it should:

1. Provide a pull down action menu, offering such options as *follow-link, compute-link, start-link,* and *end-link.*
2. Package a user selected object (such as a text span) and an action chosen from the menu, and communicate this message to Microcosm.
3. The viewer should be able to highlight certain objects, so that they may be treated as buttons.
4. It should be possible to launch the application with a given data set so that the application is shown in a specific state, such as with a particular item of data on display.

In order to achieve all the above functions, it is clear that we must be able to have access to the source code of the application, but this is not always possible. So for this reason, in Microcosm we have specified a spectrum of viewer awareness.

1. FULLY AWARE MICROCOSM VIEWERS.

These are viewers that are fully integrated with Microcosm: that is the source code of the viewer is available and has been enhanced to include calls to the Microcosm API. This open interface allows for the creation of messages according to a flexible communication protocol and therefore allows the viewer to communicate directly with Microcosm. We have written such viewers to deal with data formats that are common in our environment, such as text, bitmaps, video, audio, vector graphics and rich text. A Microcosm viewer may display active areas as buttons. These are combinations of object selection and action which are in some way highlighted, and the information about what to highlight is stored in the linkbase. The viewer requests this information when loading the data. Microcosm viewers may also be asked to start up with the focus at any particular point in the data.

2. PARTIALLY AWARE VIEWERS.

These are applications from third party sources which we have adapted to be Microcosm aware. Many packages such as Word for Windows, Toolbook and Access have some level of programmability. In such applications it is quite straightforward to write the necessary code to produce an action menu and to package a selection and action into a message for dispatch to Microcosm. The process of adapting such applications is qualitatively quite different from re-writing the application to become link-aware, and is the sort of task that may be undertaken by any competent user of a package, given appropriate guidance. Adding simple link awareness usually requires less than 100 lines of macro code. In some cases it is possible to use this method to produce a fully aware viewer, and we have achieved this with Word for Windows.

3. UNAWARE VIEWERS.

In the worst case, where it is not possible to build into the viewer any form of action menu, we have introduced the idea of "clipboard links". The user makes a selection from the application in the normal way, and then copies the selection to the clipboard. An action menu may then be chosen from the Microcosm icon, and Microcosm will then take responsibility for taking the clipboard contents and the chosen action and packaging them into a message. A refinement of this approach allows the user to order Microcosm to "monitor the clipboard": whenever the clipboard contents change Microcosm will automatically package the new contents with a pre-selected action, such as *follow-link*, or prompt the user for a specific action.

MICROCOSM IN ACTION

Microcosm has now been in use for over two years. The first applications developed were educational but Microcosm is being increasingly used for industrial applications such as training and multimedia information management. The first application integrated with Microcosm was Microsoft Word. The programmability of Word makes it an ideal application to be made fully Microcosm aware. The user can follow Microcosm links of all types from Word documents and of course any link can be created with a Word document as its destination. Word has been used very successfully by undergraduate students to create hypermedia dissertations based on Microcosm applications [Cols92]. We have also used Word to create an viewer for text documents marked up in SGML-like formats and therefore to apply Microcosm links to documents of this type. The Computing in the Humanities group at Oxford University are using this technique to build a hypertext version of the Project Electra documents.

Apart from being used as the basis for the creation of new material, the Microcosm link service can also be used to integrate existing hypermedia applications into a wider resource-base. We have used this technique successfully with material created in Toolbook [Crow92]. This enables Toolbook applications to be linked via Microcosm to large multimedia resource-bases which allows the application to be enhanced, extended or customised for different users. We have also implemented a dictionary facility that permits users to select text in a Toolbook window and follow a Microcosm 'dictionary' link via a Toolbook menu selection to a definition of the selection. A dictionary linkbase can be easily created automatically from an electronic dictionary thus providing the dictionary links with no authoring effort. This facility has been used successfully in a number of industrial training applications.

We have integrated databases and spreadsheets into Microcosm to enable users to access the data processing capabilities of these applications whilst still having access to a set of multimedia resources containing information describing or enhancing the data held in the processing system. For example, we have used Superbase to maintain a database of images from a photographic archive. A 'display' feature we have built into Superbase enables the user to display the image related to the record retrieved by the database search via a Microcosm linkbase. Microcosm links can be followed from Superbase records using the clipboard facility.

The applications of the Microcosm link service described so far have all been based on the selection of text from information displayed on the screen. However, the design of Microcosm allows any type of information to be used to initiate a link retrieval. Two examples of this extended functionality are the integration of Microcosm with AutoCAD and with the geographical information system SPANS. AutoCAD names the objects that it uses to create its 3D representations. We have used this facility as the basis for linking these objects to related information in a Microcosm application. When the user selects an AutoCAD object and the Microcosm 'follow link' action from the menu bar, the name of the object is sent to Microcosm as part of the 'follow link' message and used as the basis of link retrieval from the linkbase. In SPANS the user selects a point on a SPANS map and the 'follow link' action. The co-ordinates of that point on the map are then used to retrieve links from the linkbase. These techniques provide a seamless interface between the third-party information system and the Microcosm application. The user need not be aware that they are using Microcosm, since it is merely being used as a service by the information system to access multimedia data files (9).

THE FUTURE OF MICROCOSM

Microcosm is currently most fully implemented for PCs running Microsoft Windows but prototypes have been developed for the Macintosh and X-Windows platforms. A commercial version of Microcosm is currently under negotiation.

Research is continuing on various aspects of the system in particular the implementation of a distributed version which will allow the sharing of information between users. In particular the sharing of information between the various heterogeneous versions of Microcosm, and the incorporation of information services such as WAIS and World Wide Web.

REFERENCES

[Cols92] Colson, F. & Hall, W. "Educational Systems: Pictorial Information Systems and the Teaching Imperative." Images and Manuscripts in Historical Computing (ed. Thaller, M.), Scripta Mercaturae Verlag, 1992, pp 73-86.

[Crow92] Crowder, R. & Hall, W. "The Use of Interactive Media as a Training and Operational Interface in the Advanced Factory." Proceedings of the Third International Conference on Factory 2000, (University of York, UK, July), 1992, pp 106-110.

[Davi92] Davis H., Hall W., Heath I., Hill G. & Wilkins R., "Towards an Integrated Information Environment with Open Hypermedia Systems." Proceedings of the Fourth ACM Conference on Hypertext (Milan, Italy, November), 1992, pp 181-190.

[Hall92] Hall, W. & Simmons, D. "An Open Model for Hypermedia and its Application to Geographical Information Systems" Proceedings of EuroGraphics'92, (Cambridge, UK, September), 1992.

[Hill93] Hill G., Wilkins R., Hall W., "Open and Reconfigurable Hypermedia Systems: A Filter Based Model." Hypermedia, Vol 5, No. 2, 1993.

[Li92]. Li Z., Hall W. & Davis H., "Hypermedia links and information retrieval.", Proceedings of the 14th British Computer Society Research Colloquium on Information Retrieval, (Lancaster University, U.K.), 1992.

[Malc91] 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, December), 1991, pp 13-24.

[Pear89] Pearl A., "Sun's Link Service: A Protocol for Open Linking." Proceedings of Hypertext 89 (Pittsburgh PA, November), 1989, pp 137-146.

[Rizk92] Rizk A., Sauter L., Multicard: An Open Hypermedia System. Proceedings of the Fourth ACM Conference on Hypertext (Milan, Italy, November), 1992, pp 4-10.