From Hypertext to Active Communication/Information Systems

Frank Kappe

fkappe@iicm.tu-graz.ac.at

Institute for Information Processing
and Computer Supported New Media (IICM),
Graz University of Technology, Graz, Austria

Hermann Maurer

hmaurer@cs.aukuni.ac.nz

Department of Computer Science
University of Auckland, Auckland, New Zealand
(on leave from Graz University of Technology)

Abstract

In this paper we advocate the integration of advanced communication facilities into a distributed hypermedia system. We will first develop a set of user requirements and then come up with a simple design based on three orthogonal concepts (annotation, active collections, and active documents) that can be used to implement the required features in a certain implementation of a distributed hypermedia system (Hyper-G).

We also show how allowing to apply the three concepts to every object of the information space (as implied by the principle of orthogonality) transforms the passive information system into an "active communication/information system" in which the user is not only able to contribute to the information body, but also to let the system actively seek and deliver information according to the user's interests.

1 Introduction

Most of today's hypertext/hypermedia systems are small, stand-alone applications operating in a closed environment that have almost nothing in common with the original idea of hypertext – except for the use of the "node–link" model. For an overview of current hypertext systems you may consult [17], [19] or [21].

However, we may observe recent trends towards open, distributed, large-scale information systems based on hypermedia principles[11]. There are even some concrete and very successful implementations of "Networked Information Retrieval (NIR)" systems based on Internet (e.g., Gopher[1], WAIS[20] and WorldWideWeb[4]) that can be seen as first steps towards a distributed hypermedia information system.

At the Graz University of Technology, an ambitious hypermedia system called "Hyper-G" is currently under development. It is designed as a universal, large-scale, distributed, multi-user, hypermedia information system, similar in scope to Xanadu[18], Intermedia[9] and the NIR tools mentioned above. As a first application of Hyper-G the Graz University of Technology (among a few other institutions) operates a Hyper-G-based Campus-Wide Information System (CWIS).

It is rather obvious that such a distributed system is an ideal platform to integrate electronic mail and news services and to provide facilities for CSCW (computer supported collaborative work). As a consequence, the distinction between passive "readers" who only retrieve information and active "authors" who contribute information to the network, will blur. Every user will do both.

In this paper we will outline how communication and collaboration facilities can be integrated within a concrete implementation of a distributed hypermedia system (Hyper-G). In addition, we propose to go even a step further and make the the transition from a "passive" information system that supplies only navigation and browsing facilities to an "active communication/information system" that not only lets the user become an author of information and send it to others, but also actively seeks and delivers relevant information to its users.

2 Communication Support in Hypermedia: Requirements

This section lists some user requirements for nicely integrating services like e-mail and computer conferencing into a hypermedia system. The requirements are not presented using some requirements engineering formalism; rather, they are stated in plain English together with examples so that they can be easily understood; they describe only the user's point of view. In later sections we will then investigate how these requirements can be met and implemented within the Hyper-G framework.

Requirements:

1. It should be possible to send messages to one or more other users of the hypermedia system, whether they are on-line or not (asynchronous communication).
2. It should be possible to send messages to a group of persons that is identified by a single name (alias).
3. The message itself may be just plain text, but may also contain images, digitized speech, movie clips, spreadsheets, etc. (multimedia mail).
4. It should be possible to include arbitrary links to other objects in the message (hypertext), e.g. "Have you seen **this book**?" where 'this book' is a link to a certain dataset within the system.
5. In particular, when referring to another mail message (e.g. when answering a question), a link to that message should be created automatically (e.g. "Reply to **your message of Monday, ...**").
6. When receiving a message, it should be possible to archive it in a set of user-definable areas, and to associate user-definable keywords with the message.
7. It should be possible to archive the outgoing messages in a similar fashion.
8. The user should be able to request a mail being sent at a certain time. In particular, it should be possible to have an incoming mail sent again (e.g. "remind me next Monday").
9. It should be possible to search archived mail messages by author, subject, date and by searching the full text (if applicable), e.g. "find me messages of last week that contain 'WAIS' in either the subject or the message body, but not the ones I sent myself".
10. It should be possible to install discussion groups similar to USENET news, where a larger audience discusses certain topics.
11. It should be possible to participate in the discussion by a mechanism similar to e-mail, i.e. by sending a message to the discussion group instead of a person.

12. The system should allow an easy overview of threads of a discussion, if possible a graphical representation of questions, answers, arguments in favor or against (similar to the gIBIS[6] system).
13. A user should be able to express special interest in a certain discussion group (to "subscribe" to that group), so that new contributions to that discussion are also sent as messages to the user. The advantage for the user is that only one mailbox has to be checked, not a number of newsgroups (subscribed newsgroups are similar to Internet mailing lists).
14. However, the user should be able to filter the messages of a subscribed discussion group (e.g. "I dont want to see messages from author 'foo' and messages that do not contain the word 'foobar' in them").

3 Basic Features of Hyper-G

This section discusses some of the basic features of the Hyper-G system, as far as they are necessary to understand how the facilities defined by the above requirements can be integrated into Hyper-G. For an in-depth description see [10], [13], [12] and [14].

Hyper-G is a large distributed hypermedia system suitable for a wide range of applications. Mega-quantities of documents of different types (text, graphics, raster images, video- and audioclips) can be stored and associated with each other. This necessitates storing documents in a distributed fashion in a network consisting of potentially thousands of computers. It also implies that the usual sole reliance on links for navigation as found in most hypermedia systems has to be supported by other techniques. In addition to typed **links**, Hyper-G permits the structuring of information into hierarchies of so-called **collections** to present the user with a rough feeling of what might be found where. Hyper-G also supports advanced techniques such as fulltext and "fuzzy" **search** in domains (sets of collections) defined by the user. Further, **guided tours** are available as help given by experts to persons first entering the "information jungle" of such a large system.

Hyper-G is implemented using the client-server model. The server (in our case called the "link server") contains information concerning the documents, such as title, keywords, access privileges, date of creation etc., as well as all the links between documents. The link server itself is a distributed object-oriented database specifically developed for Hyper-G. It allows for automatic maintenance of collections, documents and links even across server boundaries, and handles complex queries over a set of collections.

Links in Hyper-G are defined between *source anchors* and *destination anchors*. This allows a link to point to specific parts of a document (e.g., a paragraph). Links are bidirectional,

i.e. the link server can find all documents with links pointing to a given document, and links are stored separately from documents (i.e. it is not necessary to modify a document in order to attach a link). This is essential for automatic maintenance of the web, for customization on a per-user basis, and also for support of annotations and discussions, as we will see.

Clients implement the user interface to Hyper-G. Currently, a client for VT100-like terminals is in used, while more advanced clients for X-Windows and MS-Windows are being developed. Hypermedia specialists are often at odds about which of the many usage-paradigms is optimal: the "stack-metaphor", "the book metaphor", "the holiday travel paradigm", or whatever else (see [8]). Hyper-G does not attempt to resolve this issue but rather supports the idea that each of the above ways of interfacing with a hypermedia system may have its advantages depending on the user, the application and the workstation. The choice of user the interface is left to the client developer, the system administrator and even the individual user.

Clients connect to exactly one Hyper-G server (the "nearest" in terms of bandwidth). Whenever information is needed from other servers the data is routed through this server (and also cached there so that it will be available faster next time). This single server is also responsible for maintenance of user data, user groups and access rights for its domain.

Hyper-G can be used in four identification modes:

1. In **identified mode** users identify themselves to their Hyper-G server by username and password. Identified users can create, modify and delete Hyper-G objects, subject to access permissions.
2. The **semi-identified mode** also allows to perform write operations on objects. The difference to identified mode is that the real identity of the user is known to the system, but to nobody else. This assures that users are prevented from making slanderous comments, contributions violating some pornography law, etc. (since they can be identified by the law, if necessary) yet their real identity is shielded from other users: they can choose an arbitrary pen-name. This mode has proven to be particularly valuable for discussions in groups to assure that the usual barriers of job-hierarchy or such are broken down, thus stimulating free discussion.
3. The **anonymously-identified mode** is similar to the the semi-identified one except that the real identity of the user is not known even to the system. Users can select any still available pen-name and password, and this combination can be used whenever accessing the system. The purpose of this mode is that user preferences, bookmarks etc. can be stored across session boundaries.
4. The **anonymous mode** can be used most of the time to avoid the "Big-Brother-Syndrome" [16]. Of course anonymous users are not allowed to enter or modify information.

In all modes with the exception of anonymous mode the user has write access to a private **home collection**, where user interface preferences, history, bookmarks etc. are stored.

4 Integration of Communication Features within Hyper-G

It was Aad van Wijngaarden [22] who introduced the notion of *"orthogonality"* into programming language design: a feature available in one context should be available in all analogous contexts. We have attempted to design Hyper-G along these lines, so that we achieve maximum functionality with a minimal set of simple concepts. In this section we describe how communication support according to the requirements of section 2 can be implemented using only three simple concepts.

The rule of orthogonality implies that these concepts should not be available for communication only, but must be available everywhere in the system. We will show how much the "information system" part of Hyper-G benefits from the new concepts when they are applied to other contexts.

4.1 Annotations, Discussions, Electronic Mail

As has been said already, Hyper-G users are not just passive consumers of information, but may actively insert information into the database. They may do this by creating a local hierarchy of collections in their home collection, and putting information in there (similar to how UNIX users may insert data into their home directories). They may also write data to collections outside their personal hierarchy if the access rights of that collection allow to do so. In addition, any document created by a user may contain links to arbitrary documents of the whole Hyper-G database.

This simple concept can be used for implementation of annotations, discussion groups and e-mail within Hyper-G:

4.1.1 Annotations

In Hyper-G, an annotation is a document that is attached to another document by a link of type "annotation". Depending on access rights of that link we may distinguish between private annotations, group annotations (accessible only by a certain user group) or public

annotations (accessible by everybody). The annotation will typically contain some kind of comment on the annotated document (e.g. "The information given here is outdated, I found **newer figures** (link to another document)".

Exactly how annotations are visualized depends on the client that is used to access the information, and on user preferences. A client may choose to always show all annotations attached to a document, or it may open only the private ones, or it may offer a button: "Show annotations"...

The annotation link is always emanating from the annotation and pointing to the annotated document. When annotating a document it is not necessary to modify that document. This allows to annotate documents without having write permission on that document. The support for bidirectional links of the Hyper-G link server allows to find both the annotated document from the annotation, and all annotations for a certain document.

The annotation can either be put into the author's home collection, or anywhere else where access permissions allow it.

4.1.2 Discussion Groups

Because any document may be annotated, annotations themselves can be annotated, possibly by a different user. In this way an electronic discussion starts. For the purpose of such a discussion, annotations (let's call them contributions now) will have to be accessible by the public or at least a user group, and it will be useful to not store them in the author's home areas, but to create a special collection for them, or better still a hierarchy of collections (like USENET news).

In addition it is possible to define more fine-grained link types (such as "counter argument", "generalization", "supporting example" like in gIBIS [3]), so that in a graphical overview of the collection the flow of discussion can be followed more easily (figure 1).

4.1.3 E-Mail

Let us assume that all users have a collection named "Mailbox" under their home collection, with write permissions for everybody and read permissions only for the user concerned. Then sending a message to that user can be performed by creating a document and "sending" it to the recipient's mailbox. The "send" operation is actually just the association of that document with the receiver's mailbox collection. It should be noted that

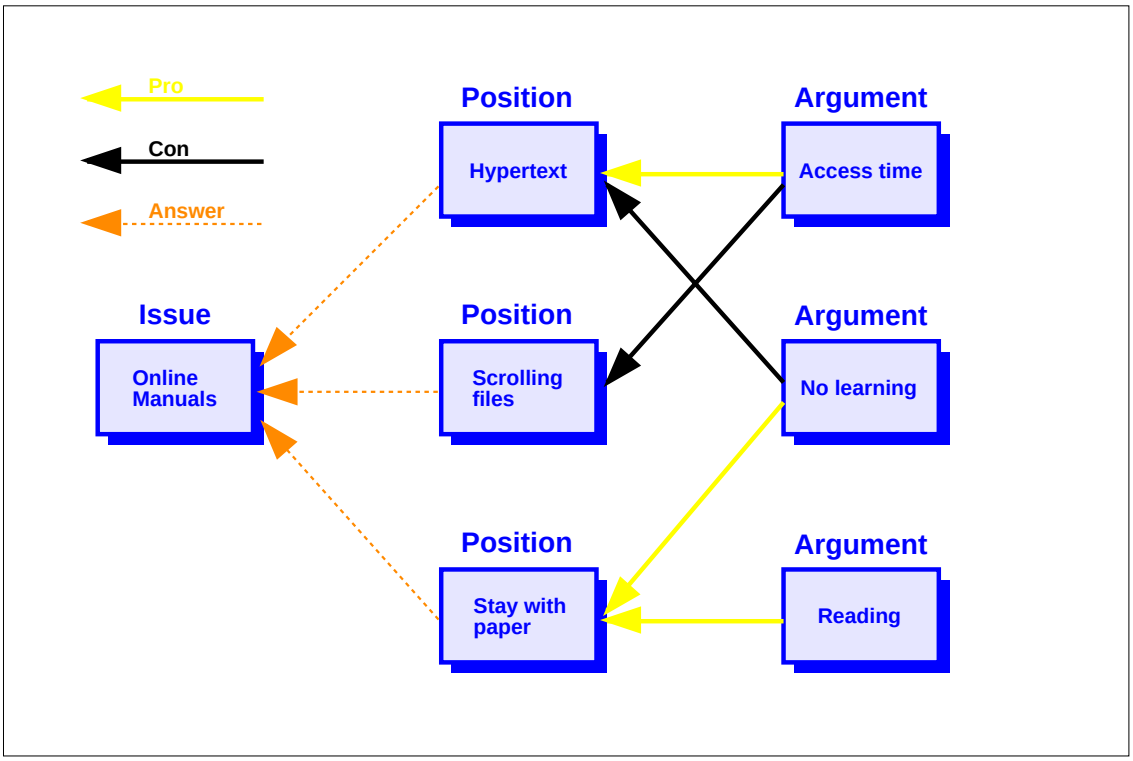

Figure 1: Overview of a discussion in gIBIS

one document can be associated with more than one collection, so there is no need to actually copy the document, even if it is sent to a number of users. The sender may also send the document to one of the user groups defined in Hyper-G[1], meaning that it is sent to each member of that user group (see requirement 2).

The client may (or may not) treat the mailbox collection special in that it periodically checks whether it contains new documents. Then the user may take the document out of the mailbox and put it into any other collection (actually, even into more than one collection) in the home area ("archive" it, requirement 6), or simply remove it. Similar, the sender may also archive the document in one of the private collections belonging to the sender (requirement 7).

Let us now consider the case where the recipient of a mail wants to comment on it (say, answer a question). It is easy to see that this again is nothing else than an annotation. The link from the answer to the question will be automatically created as with any annotation (requirement 5), and because of Hyper-G's bidirectional links it is easy to find

[1]Hyper-G users are organized in a hierarchical group structure, so that it is easy to specify meaningful access rights like "readable by all members of Graz University".

the question from the answer, and vice versa (it is a typical problem of newsgroups and mailing lists that it is usually easy to find the question when you have the answer, but difficult to find all answers to a certain question).

It should be noted in passing that whenever we say "document", we mean not only text, but all document types the client can handle (like images, sounds, animation, etc.). You can, for instance, speak a question into a voice document, send it to a number of people, and later on collect the (spoken) answers of the people. This adds to Hyper-G a functionality similar to an answering machine.

4.2 Active Collections

We will now introduce the concept of **active collections** in Hyper-G: Users may specify their interest in certain areas of the information space of Hyper-G. This is done by associating a query with a collection. The collection becomes "active" in that the user's query is checked against every document inserted or modified within that collection (and its subcollections, recursively). If the document matches the query it is sent as a mail to the user. As a special case, an empty query always matches.

Hyper-G's query mechanism is rather powerful: It allows boolean combinations of queries by attributes such as author, creation/modification date, keywords, and title of the document as well as fuzzy full text searches, combined with relevance feedback ("document should be similar to this one"). According to the orthogonality rule the full set of query mechanisms is also applied to the filtering process in active collections.

It is rather obvious that this concept can be used to implement the subscription to discussion groups demanded in requirements 13 and 14 when it is applied to a collection that hosts an electronic discussion. Activating such a discussion results in transforming the "passive discussion" (the user has to check whether some new contributions were made, like in the USENET news) to an "active" one (the user is informed by the system when new contributions are entered, like in Internet mailing lists).

One advantage for the user is that using subscribed discussion groups only one mailbox has to be checked, and even that can be done periodically by the client software without the user's manual intervention. A second advantage is that Hyper-G's advanced query mechanism can be used to pre-filter relevant contributions from junk (e.g. by queries like "must not be from author 'xxx' and must not contain the words 'xxx' or 'yyy' in the title and must contain either 'aaa' or 'bbb' or 'ccc' in the text").

In addition, however, the rule of orthogonality demands that this concept be applicable not only to collections that host discussions, but to *any* collection throughout the system.

For example, a user of a Hyper-G based CWIS could also subscribe to the collections "Calendar of Events" and "What's new on Campus" and automatically receive mail when something new is inserted there. A user interested in, e.g., fractals may activate the whole collection "Library" with "title or text contains the word *fractal*" as a query and receive mail whenever a new book on the topic becomes available. The user may even activate the CWIS's root collection together with the root collections of some other Hyper-G servers with the same query and be informed whenever *anything* on fractals is entered into *any* of the selected servers in *any* collection (including all discussion groups, of course).

It is easy to see how this single feature adds a new quality to a hypermedia system. It is transformed from a passive information system in which users have to struggle to find information by *navigating* and *browsing* through the information space into an "active" system in which users may passively relax and are automatically informed when information relevant to them becomes available. Of course, this is just an option; Hyper-G may also be used in the traditional way as its navigation facilities and other features remain available.

Concerning the actual implementation, there are a number of possibilities of when the queries are evaluated: It is not absolutely necessary to really match all new documents against all user queries when they are inserted. Rather, one could search for all documents matching a certain user's queries and newer than "time-of-last-search" whenever the user starts a session, or whenever the user's mailbox is checked. The latter option would require no code in the Hyper-G server at all, as the searches could be performed by the client's request just like ordinary Hyper-G searches. However, the first option seems to be more efficient, requires no code in the clients as the searches are performed automatically by the server, and also allows for the additional features described in section 5.

4.3 Active Documents

A number of attributes can be associated with Hyper-G objects (and therefore documents), among them time-related attributes like creation time, modification time, opening time (before that the document is invisible except for the author), and expiration time (after that the document becomes invisible again). We now propose to associate a "mail at/to" attribute that takes a pair of (time, user) as value. The idea is that when the specified time is reached, the document is mailed to the specified user. In analogy with active collections that send mail whenever a document is inserted into them, we will call documents with the "mail at/to" attribute **active documents**.

According to our rule of orthogonality we will allow the 'user' attribute value not only to be a single user, but also a user group (compare requirement 2). The 'time' attribute

value should not be restricted to a single time value, but rather a more flexible scheme using wildcards similar to the UNIX /usr/lib/crontab format would be useful. Such a format would allow to specify requests like "send this document every Weekday at 18:00" or "send it every 3 days".

This feature allows the delayed sending of mail messages ("send this out tomorrow at 9:00"). As a special case, the recipient of a message could move the message to some archive area, but in addition specify that it should be re-sent to the recipient at a certain time (see requirement 8). This is useful when one cannot make use of the message right now, but will need to remember it some time in the future (e.g., announcement of a meeting).

Again, applying this concept to all Hyper-G documents yields additional features unrelated to communication. It allows one to use Hyper-G as a reminder service (by sending delayed mail to yourself), and even as a personal calendar. Again, documents sent need not be text. For instance, you may digitize the sentence "time to go home" and request it being sent every working day at 6 pm. Depending on the capabilities of the client platform and the client software, messages may be displayed/spoken automatically when received.

4.4 Implementation details

4.4.1 Access Rights

Of course, Hyper-G's implementation of access permissions and identification modes will have to be reconsidered in the light of the new communication functionality.

Identification Level	anonymous	anon-identified	semi-identified	identified
Send mail to other user	No	No	Yes[1]	Yes
Send Reminder to oneself	No	Yes[2]	Yes	Yes
Receive mail	No	Yes[2]	Yes	Yes
Write to discussion	No	No	Yes[3]	Yes[3]
Read discussion	Yes[3]	Yes[3]	Yes[3]	Yes[3]
Subscribe to collection	No	Yes[2]	Yes	Yes
Private Annotation	No	Yes[2]	Yes	Yes
Public Annotation	No	No	Yes	Yes

Table 1: Communication Facilities possible for different identification modes

Table 1 summarizes the communication possibilities allowed on individual identification levels (described in section 3). A few notes seem to be in order:

1. It is not clear whether semi-identified users should be allowed to send mail to others, as the real identity of semi-identified users is known to the system, but not to other users. That means that one can receive mail from an unknown person, which may sometimes be embarrassing. On the other hand, this is also true for current e-mail over Internet, as the mail transfer protocol is not secure. For the time being, we allow semi-identified users to send mail to others, but perhaps this may change in the future.
2. Anonymously identified users are allowed to write private data to their home collection (up to a certain quota), but only that kind of information. Therefore, anonymously identified users may not send mail or write contributions to discussions but may write private annotations and write mail to themselves, i.e. send reminders. However, they may receive mail from (semi-)identified users and the system itself and consequently may subscribe to collections.
3. Read and write access to discussion groups are subject to access permissions of that group. Of course it is possible to create discussion groups with access granted only to a certain user group.

When a document is sent as mail, (i.e. copied to the mailbox collections of the recipients), the sender remains the owner of the document but loses the right to modify/delete the document (of course, senders may choose to remove the document from their archives, but this does not mean physical deletion). Otherwise, the document could be deleted from the recipients' archives or modified after it was read.

4.4.2 Gateway to Standard E-Mail

Since we cannot expect all of today's estimated 15 million Internet users [15] to switch to Hyper-G as their favorite communications medium at once, it will be useful to implement gateways to the existing mail standard on Internet [7] so that it is possible to send Hyper-G mail to any Internet user and to receive Internet mail in Hyper-G.

While the outbound mail (i.e. mail sent from Hyper-G) is already in the core functionality of Hyper-G, the reverse direction requires implementation of a mail gateway. Users can direct the mail system to forward their mail to this gateway which in turn copies it into the users' mailbox collections. The demanding part is the automatic construction of links between questions and answers. In most cases, however, it can be done using the unique message ID associated with every internet mail message.

Of course, multimedia mail requires special treatment. There are some platform-dependent ways to transmit multimedia mail (e.g. NeXTmail, SUN-attachments), but we would prefer to encode multimedia mail to and to parse it from MIME (Multipurpose Internet Mail Extension) as standardized by Internet RFC 1341 [5].

4.4.3 Gateway to USENET News

Similar to the mail gateway it would be nice to have gateways to USENET news, one in each direction. While the outbound gateway is again trivial, the inbound gateway is not because of the need to reconstruct the question-answer relations (or the threads of the discussion) from the news. However, some work in this direction has already been performed at the Technical University of Denmark [2].

The advantage of mirroring selected newsgroups in Hyper-G is that users may subscribe to them, including the use of the advanced filtering mechanisms provided by Hyper-G. Also, Hyper-G's graphical collection browser will allow to follow the discussion more easily (see figure 1).

5 Active Communication/Information Systems

So far we have shown how the following features not directly related to communication provide a tremendous increase of functionality by applying the principle of orthogonality to the concepts of annotation, active collections and active documents wherever possible:

- The concept of annotation should be applied to and allowed for any document in the system. It is a powerful way of customizing the view of the information space.
- The concept of active collections when applied not only to discussion groups, but to any collection in the system allows to transform the otherwise passive information system into an active one that actively searches for new contributions interesting for its users.
- The concept of active documents allows the user to use the system as a reminder service and even as an electronic time schedule.

Especially the notion of an "Active Communication/Information System" (let's use the acronym ACIS from now on) deserves a few general comments. It is a step from the usual passive system that lets the user do the searching towards an intelligent servant that does the searching for the user. Of course, since even the most sophisticated query mechanisms are not intelligent enough to really understand the text they are scanning, a hybrid system that allows to automate the rather simple tasks ("anything new about fractals in the library?") and let the user concentrate on finding answers to the more complicated questions would be a feasible solution.

We could even push the notion of an ACIS a bit further: What if we associate priorities with mail messages and therefore also with messages generated automatically (e.g, based on the ranking when comparing incoming messages with a set of user-defined keywords)?

This would make it feasible to implement a simple mail robot that sends us a fax or even gives us a phone call in the middle of the night when something very important (i.e. something with a high priority ranking) for us was received in the mailbox (it could even read the message through the phone using speech synthesis). E.g., a stock broker may wish to subscribe to a certain commercial news service and specify to be immediately informed of any news on a certain company. Now this would really be an active communication/information system!

However, as tempting as it may seem to totally rely on an intelligent agent to find the things we want for us, there are also some dangers in it. It potentially narrows our view of the universe as we would only see information that we have expressed interest in beforehand, i.e. the "browsing" phenomenon when using printed information would disappear: we would never see some other interesting information that could be found on the opposite page of a paper-based newspaper. Scientists will not notice the development of new disciplines of research if they read only articles that match their core interests.

It has been suggested that the interests of users need not be specified by the users themselves, but by monitoring the kind of information they looks at and presenting similar information. Of course, this raises the issue of privacy because the system easily gets an overview of the users' reading habits and interests. However, the issue also exists in traditional mailing lists and bulletin boards. How can we make sure that the information that a user has subscribed to, e.g., a discussion group on homosexuality, cannot be used against the user?

6 Conclusion

From the original idea to augment the capabilities of a hypermedia-based information system with communication facilities we have developed the notion of an Active Communication/Information System (ACIS). Based on previous experience with large-scale information systems like videotex we believe that a successful information system can not just present information to a user community willing to search for it. Rather, we think that the next-generation information system will be a large-scale, distributed, multi-author, multi-purpose hypermedia system with increased emphasis on communication features. In addition, it will be an active system in that it behaves like an intelligent servant that relieves us from the annoying tasks of looking for the same kind of information in the same places again and again, and lets us concentrate on the more difficult tasks of finding information that we dont know (yet) that it exists and that it is interesting for us.

References

[1] Alberti B., Anklesaria F., Lindner P., McCahill M., and Torrey D.: "The Internet Gopher Protocol: A Distributed Document Search and Retrieval Protocol". March 1992. Available by anonymous ftp from boombox.micro.umn.edu in directory pub/gopher/gopher_protocol.

[2] Andersen M. H., Nielsen J., and Rasmussen H.: "A Similarity-based Hypertext Browser for Reading the UNIX Network News". *Hypermedia*, 1(3):255–265, 1989.

[3] Begeman M. L. and Conklin J.: "The Right Tool for the Job". *Byte*, 13(10):255–266, October 1988.

[4] Berners-Lee T., Cailliau R., Groff J., and Pollermann B.: "World-WideWeb: The Information Universe". *Electronic Networking: Research, Applications and Policy*, 1(2):52–58, Spring 1992.

[5] Borenstein N. and Freed N.: "MIME (Multipurpose Internet Mail Extensions), Internet RFC 1341". June 1992. Available by anonymous ftp from nic.ddn.mil, file rfc/rfc1341.txt.

[6] Conklin J. and Begeman M. L.: "gIBIS: A Hypertext Tool for Exploratory Policy Discussion". *ACM Transactions on Office Information Systems*, 6(4):303–331, 1988.

[7] Crocker D. H.: "Standard for the format of Internet text messages". August 1982. Available by anonymous ftp from nic.ddn.mil, file rfc/rfc822.txt.

[8] Davies G., Maurer H., and Preece J.: "Presentation Metaphors for Very Large Hypermedia Systems". *Journal of Microcomputer Applications*, 14(2):105–116, April 1991.

[9] Haan B. J., Kahn P., Riley V. A., Coombs J. H., and Meyrowitz N. K.: "IRIS Hypermedia Services". *Communications of the ACM*, 35(1):36–51, January 1992.

[10] Kappe F.: *Aspects of a Modern Multi-Media Information System.* PhD thesis, Graz University of Technology, Austria, June 1991. Also available as IIG Report 308; IIG, Graz University of Technology (Jun 1991), and by anonymous ftp from iicm.tu-graz.ac.at in directory pub/Hyper-G/doc.

[11] Kappe F.: "New Aspects of Large-Scale Hypermedia Systems". In *Proc. Hypermedia in Vaasa '93*, May 1993.

[12] Kappe F. and Maurer H.: "Hyper-G: A Large Universal Hypermedia System and Some Spin-Offs". *ACM Computer Graphics, experimental special online issue*, May 1993. Available by anonymous ftp from siggraph.org in directory publications/May_93_online/Kappe.Maurer.

[13] Kappe F., Maurer H., and Sherbakov N.: "Hyper-G – A Universal Hypermedia System". *Journal of Educational Multimedia and Hypermedia*, 2(1):39–66, 1993.

[14] Kappe F., Pani G., and Schnabel F.: "The Architecture of a Massively Distributed Hypermedia System". *Electronic Networking: Research, Applications and Policy*, 1993. To appear.

[15] Lottor M.: "RFC 1296: Internet Growth (1981-1991)". January 1992. This report, as well as up-to-date data is available by anonymous ftp from ftp.nisc.sri.com in directory pub/zone.

[16] Maurer H., Roszenich N., and Sebestyen I.: "Videotex without Big Brother". *Electronic Publishing Review*, 4:201–214, 1984.

[17] Maurer H. and Tomek I.: "Hypermedia – From the Past to the Future". In *Proc. New Results and New Trends in Computer Science*, pages 320–336, LNCS 555, Springer Pub. Co., 1991.

[18] Nelson T. H.: *Literary Machines, Edition 87.1.* The Distributors, South Bend, IN, 1987.

[19] Nielsen J.: *Hypertext & Hypermedia.* Academic Press, San Diego, CA, 1990.

[20] Stein R. M.: "Browsing Through Terabytes – Wide-area Information Servers Open a New Frontier in Personal and Corporate Information Services". *Byte*, 16(5):157–164, May 1991.

[21] Tomek I., Khan S., Muldner T., Nassar M., Novak G., and Proszynski P.: "Hypermedia – Introduction and Survey". *Journal of Microcomputer Applications*, 14(2):63–100, 1991.

[22] van Wijngaarden A.: *Orthogonal Design and Description of a Formal Language.* MR 76, Mathematisch Centrum, Amsterdam, October 1965.