
Semantic control of open hypertext systems by typed objects

Rainer Hammwöhner and Rainer Kuhlen

Department of Information Science, University of Constance, Germany

Received 6 October 1993

Abstract. The C(K)onstance Hypertext system (KHS) is an elaborated example of an open hypertext system. Open hypertext systems combine heterogeneous resources and different hypertext applications such as e-mail, text and image hypertexts with access to online databases or other information resources. These hypertexts, to which heterogeneous users have access, grow steadily in size. This paper takes an object-oriented approach and proposes a rigorous typing of hypertext objects to overcome the problems of open hypertext. The advantage of this typing is shown on an e-mail hypertext as a special application of the KHS system. KHS is written in Smalltalk and is being developed on UNIX-machines.

1. Open hypertext – a major issue for hypertext research

The concept of an open hypertext system [2, 7] is of central importance in the overlapping fields of information science and hypertext. By open hypertext systems, we have in mind systems which:

- (1) are used (simultaneously) by heterogeneous users (for example, members in a research unit) with different read/write permissions and private spaces;
- (2) grow steadily with respect to both size and interconnectability;

- (3) combine different hypertext applications derived from the same hypertext model [12];
- (4) allow access to, and the integration of, external information resources such as e-mail, online databanks, text and image processing, or any such applications available on the host operating system;
- (5) make possible different user-specific views of the global hypertext base.

The opening of hypertext to an arbitrary number of users, to many different applications and to the use of external information sources such as online retrieval systems, is of major impact on the use and administration of hypertext. The problems which will arise can be characterised as the following:

- (1) the quantity and diversity of information may intensify the disorientation problems already known from conventional hypertexts;
- (2) one standardised interaction strategy for browsing and exploration may not be sufficient for all applications which are to be supported by the hypertext;
- (3) some applications need access to structured data, e.g. the address of a person;
- (4) new applications need to be permanently integrated in both a semantically controlled and efficient way;
- (5) external information sources often do not present their data in ways in which the hypertext applications are designed to handle. Therefore, transformation processes have to take place which are specific to the type and source of information.

2. Typing of hypertext objects – a prerequisite for open hypertexts

Before we show how to overcome these problems by typing hypertext objects, a short remark on our understanding of typing may be necessary, in particular to

Correspondence to: Professor R. Kuhlen, Department of Information Science, University of Constance, Box 5560, D-78434 Konstanz, Germany. Tel: +49-7531-882879. Fax: +49-7531-882601. E-mail: (hamm, kuhlen)@inf-wiss.uni-konstanz.de (Internet).

avoid the confusion which may arise when, for example, the terms 'typed links' and 'labelled links' are used as synonyms (e.g. [6]). Our notion of types is derived from the concept of classes as implemented in object-oriented languages and enhanced by a mechanism of dynamic type checking. Labelled links only indicate their discursive function by name, whereas link types define that function by controlling type and content of the connected units.

The features of object-oriented typing relate to the problems mentioned above in the following way:

- (1) types define restrictions on the possible content of hypertext objects and their combination. Thus, authors receive support in the construction of consistent hypertexts, and readers can anticipate what kind of information they will find inside or in the context of a hypertext object;
- (2) type definitions can also be used when hypertext structures are created automatically by the system;
- (3) inheritance as provided by a type system is a means of deriving new application oriented hypertext object types from existing ones in an efficient way.

We will discuss these features in detail in the following Sections, with special attention to the typing of hypertext units and links (and hypertexts as wholes), and refer this discussion to the C(K)onstance Hypertext System (KHS) which is being developed in the Department of Information Science at the University of Constance. Our discussion of hypertext object types will be close to this hypertext model. Because of the basically object-oriented approach of KHS, all of these objects share generic properties. The following features are therefore common to all KHS objects, and they are shared by all KHS hypertext objects.

Each KHS object knows:

- (1) by whom it has been created (modified);
- (2) whether it has been locked for editing – a prerequisite for multi-user access to the hypertext;
- (3) which user or group of users has read or write permission;
- (4) whether it is part of the hypertext permanently or temporarily (for the current session only);
- (5) how it will be presented in the user interface;
- (6) how it will be represented for the purpose of permanent data storage.

Some of these features are essential for open hypertexts, for instance, the multi-user capability, but do not enhance the expressive power of the hypertext model. The generic features of access control, object persistence control and presentation management will be refined in response to the demands of the respective subtypes.

On the topmost level of the inheritance hierarchy, hypertext objects are classified into units, links and hypertexts, object types which are dealt with in the following three Sections of this article.

3. Typing of units

Hypertext units are devoted to the representation of the content information of a hypertext, whereas links realise the relations between such items of information. Typing of units means imposing restrictions on their internal structure, presentation and interactive behaviour. From a formal point of view, KHS hypertext units can be classified most basically into composite units and media units.

3.1. Composite units

The shortcomings of a mere node/link structure of hypertext [13] has led to the concept of controlled hypertext objects, in particular nodes – so called composite units – which may contain nodes themselves¹. The KHS hypertext model regards composite nodes as the backbone of the hypertext structure and as a means for structured navigation. Any unit within the hypertext – with the exception of only a couple of so-called 'top level units' – must be part of at least one composite unit. In practice, it will be part of several units and, thus, polyhierarchically arranged. This polyhierarchy of composite units has the following advantages when users navigate through the hypertext:

- (1) the position of the unit actually visited within the structure hierarchy is presented to the user in a special window, thus providing orientation;
- (2) the subunits of a composite unit are ordered, thus providing a basic form of path mechanism which can be followed sequentially;
- (3) the presentation of links may depend on the structural embedding of their origin. A rather similar approach is proposed in [5]. If the corresponding unit, or at least one of its superordinates, is contained in more than one composite unit, the presentation of a link may be restricted to one of these alternatives (see Section 5.2);
- (4) the types of units which are allowed in a composite unit are subject to type checking. If users know what kind of composite unit they have entered,

1. Early hypertext-systems which allowed composition are Textnet [22] and Augment [8].

they will also know what kind of subunits may be contained in it.

Subtypes may be established by redefining types of subunits with corresponding ordering criteria. The following kinds of unit ordering are already in use:

- ordering by an author defined sequence;
- ordering by alphabetical name;
- ordering by creation time.

Composite nodes serve structuring purposes only, and cannot contain any media information, with one exception. *Sound* can be attached to any unit, thus providing an additional channel for information transfer. Sound attached to composite units may give meta information about how to proceed in hypertext navigation. Composite units can also be indexed, and the index terms are a further navigational and orientation means.

3.2. Media units

The leaves of the structure hierarchy tree contain the media information. Each terminal unit may (and should) contain text which provides the main information or explains the function of additional media within the hypertext discourse. Subtypes of media units specify what kind of media are available and how they are presented. There are four main subtypes of media units.

- (1) Text units contain text only.
- (2) Image units contain pictorial information in addition.
- (3) External application units contain formatted data which may be processed (and visualised) by an external application programme, for instance, vector graphics, postscript graphics, etc. This kind of unit therefore allows the integration of several external tools, an important prerequisite of open hypertext-systems [7].
- (4) Form units allow targeted access to (semi-)structured data [18] and allow specific automatic generation of links, e.g. from attribute to attribute. Because they have slots which contain one or more entries, they are similar to frames, which are well known in knowledge representation. Number and type of slot and the corresponding access methods are defined by the creation of subtypes of 'form unit'. Form units are the main resource for computation running on the hypertext, and most of the formatted data included from external information sources will be represented as form units, for instance, addresses or bibliographic units from online databases.

4. Typing of links

Imposing appropriate links on a set of distinct discourse elements may transform the set into a cohesive interconnected discourse. The quantity of virtually possible connections between two (or more) units (hypertext nodes and links) requires an explicit differentiation of link types according to their function in the discourse.

4.1. Extensional vs. intensional definition of links

In analogy to relations and sets, links may be defined by their extension or their intension. In the first case, any link between two or more units must be defined explicitly (intellectually) by an author. No rules for assigning links to new units may be derived. In the second case, semantic rules sufficiently define the conditions under which a link of a certain kind may connect some units. The selection of links may be supported by filtering processes, thus excluding link types from the presentation which – with respect to some user model or profile – are known to be of no interest to the reader.

Both of these approaches have their merits and demerits.

- (1) Extensional definition of links requires the author's attention to every single link which is to be entered into the hypertext. Because it is virtually impossible to exhaust all possibilities for creating links, even in fairly small hypertexts, and because the intellectual creation of links is a sophisticated and tedious selection process, extensional links need to be supplemented by (automatisable) intensional ones.
- (2) Intensional definition of links requires the specification of computable rules. The decision as to whether some units are interlinked may be left to the hypertext system, which will exhaustively specify all linking possibilities. The expressive power of intensionally defined links depends on the content of the corresponding rules.

Because nobody has yet succeeded in defining sufficient (and automatisable) preconditions for rhetorical relations, hypertext systems designed for authoring depend on the extensional definition of links. On the other hand, hypertexts which have to cope with large quantities of data must rely on link intension. In most cases, these links are defined on the basis of term frequencies and association factors, respectively [21, 10].

The hypertext model of KHS makes a compromise in

the question of link definition. A set of well controlled link types, such as textual reference (cf. [11]), reference to glossaries, association based on term frequency (cf. [3,16]) are defined intensionally. If they can be computed in reasonable time, they are created at navigation time and, therefore, exist only temporarily (temporary links). Otherwise, they are computed at the time of the next hypertext update and then stored permanently (permanent links). When the hypertext author has the responsibility for the creation of links, the system can give some assistance:

- (1) knowing the units to be interlinked the system makes recommendations as to which type to choose;
- (2) knowing the link type and one unit, the system can suggest the destination or the respective origin of a binary link.

4.2. Binary links

Binary links are the most frequently used links in KHS. Binary links are directed, and they lead from a hypertext unit or a span of information within a unit (a piece of text or some area in a bitmap image) to another unit or a span of information within it. Binary links are unidirectional, but if two link types are defined as inverse to each other, instances of these link types will be created as a pair automatically.

All types of binary links share some basic conditions which have to be checked.

- (1) There may be no doubles.
- (2) A unit may not be linked to itself.
- (3) Hotwords (-areas) must be represented in the media information.

The first two conditions must be checked at creation time, the third may be checked at any time when the content of the connected unit has been edited. Subtypes may employ more specific integrity rules concerning the types of source and destination unit, index terms assigned to these units and the link's hotword (cf. [15]). Associative links between hypertext units which do not allow the definition of more specific integrity rules may be entered as labelled links into the hypertext.

4.3. Set oriented links

Set oriented links interconnect sets of hypertext objects. The expressive power of this kind of link is rather similar to that of composite units, but there are still some major differences in their use. Composite units may be thought of as a reification of relational

information. The structure hierarchy is a landmark in hypertext navigation; it must not be modified temporarily. Set oriented links are used for the clustering of hypertext units due to questions of transient interest. Subtypes of set oriented links are defined intensionally and created by the system. They implement a combination of search profiles, which are known from information retrieval, and the browsing paradigm of interaction.

5. Tailoring of hypertexts to user needs and application regularities

5.1. Typing of hypertexts

The typing of hypertext may sound like a strange idea within the concept of open hypertexts. But, since, on the basis of the one KHS hypertext model, an arbitrary number of application hypertexts can be created, typing of hypertext is useful to tailor hypertext to one or several specific application domains. The following properties of hypertexts can be refined during this tailoring process:

- (1) types of units and links which can occur in the hypertext;
- (2) number and quality of active services provided by the hypertext system;
- (3) a template of units and links which is instantiated and partially filled when the hypertext is created (cf. [4]);
- (4) tools available (and necessary) to interact with the hypertext.

The necessity of providing a hypertext system with active services results from the open character of hypertext, including the use of external resources, which may produce asynchronous events. A hypertext system devoted to the management of e-mail, for instance, to be described in more detail in the following Section, must process this mail at the time when it arrives and present the retrieved mail to its reader actively.

Application-independent object types

There are several types of hypertext units and links which support tasks common to all types of hypertext.

- (1) A glossary is a composite unit containing glossary items, text units which provide short definitions of frequently used but not well-known words. Any word of the hypertext can be connected to a glossary item by a glossary link (which will be computed automatically).

- (2) Annotations contain comments on the hypertext. Any span of text can be connected to an annotation by an annotation link.
- (3) Each user has access to a private workspace, where new or deleted units can be stored temporarily.

5.2. User or group-specific nodes and links

The mechanisms mentioned above help to control large numbers of hypertext elements, but they do not reduce these data to the relevant information in the case of a real 'query' situation. KHS provides several author and user-controlled filtering mechanisms which help to overcome this problem.

Access control

The creator of any hypertext object can specify under which circumstances read/write permission may be restricted. This may be desirable, for instance, if:

- the author wants to guarantee the privacy of some informational items;
- the author wishes to restrict the access to hypertext objects for pedagogical reasons.

In both cases, readers will be wholly unaware of the existence of an object they are not allowed to read.

Context binding

Where access restriction can be used to tailor the hypertext to the demands of individual users or user groups, context binding reflects different reading situations. The same piece of text – for instance, in a unit about the essential features of modern hypertext systems – can be included in a composite unit dealing with hypertext software technology, and in another one reflecting the literary use of hypertexts as well. Depending on the context, rather different connotations and associations will be evoked. Links leading to technical papers should be seen only in the technical context, whereas links leading to remarks on hypertextual aspects of, for example, Borges' work are relevant to the literary context only.

Selection of object types

The reader of a hypertext can also control the amount of visible information by deliberately excluding link types from presentation. For example, an expert may perhaps wish to abandon the presentation of glossary links, thus reducing the risk of informational overload and saving the execution time necessary to compute

them. The information about the link types a user wants to use and see is kept in a user profile, and can be used in further hypertext sessions (see Fig. 3).

6. Shared e-mail – an example for a KHS application

6.1. E-mail as a hypertext application

E-mail has become an extensively used medium of information transfer [20] and exchange, and is well supported by modern user interfaces such as Sun's Open Windows. The mail intrinsic restriction to ASCII code (Unix standard) is no longer visible to the user, because adequate conversion routines are activated automatically. Nevertheless, some of the shortcomings which restrict the implementation of Unix mail are not yet overcome.

- (1) E-mail is still tailored to the private use of single users (but see [19]).
- (2) Archiving of mail is based on collections of flat files without interlinking or even cross search capabilities.
- (3) Mail folders store their contents permanently, but we know that most mail items are of rather transient importance, so that some kind of garbage collection would be desirable.
- (4) Mail is for information purposes only. Yet mail often contains some kind of a 'to do' list. It should be possible to assign a date to a mail unit, at which time it will pop up again to remind the user what still needs to be done.

Our scenario in developing KHS is that of a group of scientists who manage their publications, memos, etc, by means of hypertext. These scientists, working in the same department on related matters, will very often get the same mail messages from external partners or mailboxes. Other items of mail are forwarded to colleagues, and, of course, some are kept privately. Taking this situation into consideration, a shared mail archive seems to be desirable, and will be described below as an example for the integration of internet services into hypertext – which will be of increasing importance in the near future [9] – based on typed units [1].

To us, the following features of a hypertext oriented mail system are indispensable.

- (1) Incoming mail is read and converted into a hypertext unit automatically (cf. [17]).
- (2) The structural items of each mail unit (author, subject) are accessible as properties of this unit.

- (3) Each mail unit is classified automatically according to these properties. The classification leads to the insertion of the unit into one or different mail folders.
- (4) The privacy of mail must be guaranteed.

The following Sections will describe how these features are implemented by KHS.

6.2. Archiving of mail

Structure of the archive

The pivot of a KHS mail hypertext (a special KHS application derived from the same underlying model) is the archive, a system of mail folders built by special composite units, which are organised hierarchically. Generic mail folders contain subfolders, whereas special mail folders are reserved to sent mail and received mail. Index terms which are assigned to the folders describe the content of each mail to be included (cf. Fig. 1). Each mail folder is owned by a hypertext user. He is allowed to assign a protection mode to this folder, indicating who may read or modify it.

Mail folders keep the included mail only for a certain span of time, which may be assigned to each folder individually. Afterwards, the mail item will be deleted from the folder automatically. The shortest possible interval is one day, the longest one leads to permanent archiving.

Processing of arriving mail

When the mail hypertext is active (normally as a permanent background process), an active process is the watching for arriving (incoming) mail (see Fig. 1). The arriving of mail is asynchronous to other activities of the user. Therefore, even a single-user mail hypertext needs locking facilities, which prevent a unit from being updated by two processes at the same time. New mail will be read and parsed into a form unit, filling the unit slots with the mail's author, message-id, recipients, subject, text, etc. Based on the message-id, the system checks whether the mail is already included in the archive. This can happen if it has been sent to another user of the archive as well. If this is not the case, the mail's subject and the name of its sender are used as index terms, which are matched against the terms assigned to those mail folders the user is allowed to write in. The mail item is then included in those mail folders which fit best. If the matching folders differ in their protection modes, only those folders with the most private access mode are taken into account

so that the privacy of the mail is guaranteed. Every user owns a mail folder, readable only by himself, which accepts any mail not matching some other folder.

Special treatment is accorded to included mail and attached files. Included mail is cut out of the mail's text, and parsed independently into a separate mail unit, which is connected to the mail unit from which it originates by an 'included mail' link. The processing of attached files requires checking the type of included data, decoding and decompression so that a media unit representing these data can be created. At the present time, KHS accepts text, bitmaps, postscript graphics, and files as used by the Island-Tools as attached files.

Sending mail

Sending mail requires filling in a form with the mail's recipient, subject, etc. The recipient can be defined by a valid Internet address or by entering an alias. The resolving of aliases is not achieved by using the Unix mail aliases, but by reference to an archive of addresses contained in the mail hypertext. Addresses are represented by form units containing the postal and e-mail address, telephone and fax number, etc. This can be used not only for e-mail applications but also can support more traditional postal mailing.

Sent mail is classified the same way as received mail. The major difference is that sent mail which does not match any folder is simply forgotten after it is sent. Separate units which are connected by special links to the unit are included as included mail or attached files before the mail is sent. Encoding procedures take place if necessary.

6.3. Interacting with a KHS mail hypertext

To this point, we have discussed the hypertext model of KHS and a specialisation of the model designed for the archiving of e-mail. In the following Section, we will show how the features introduced above may be used interactively by a user, based on the illustration of an example dialogue. To give a full account of the interaction model of KHS would exceed the scope of this article.

The first contact between a user and KHS is by starting KHS from the Smalltalk system menu (see Fig. 2). KHS presents a launcher containing a list of hypertexts from which the user may select and load one (or more in parallel). Additionally, some information about the hypertext – type, number of terminal, composite units, etc., is presented. Thereafter, the KHS browser will be started. Supposing the user has loaded a mail

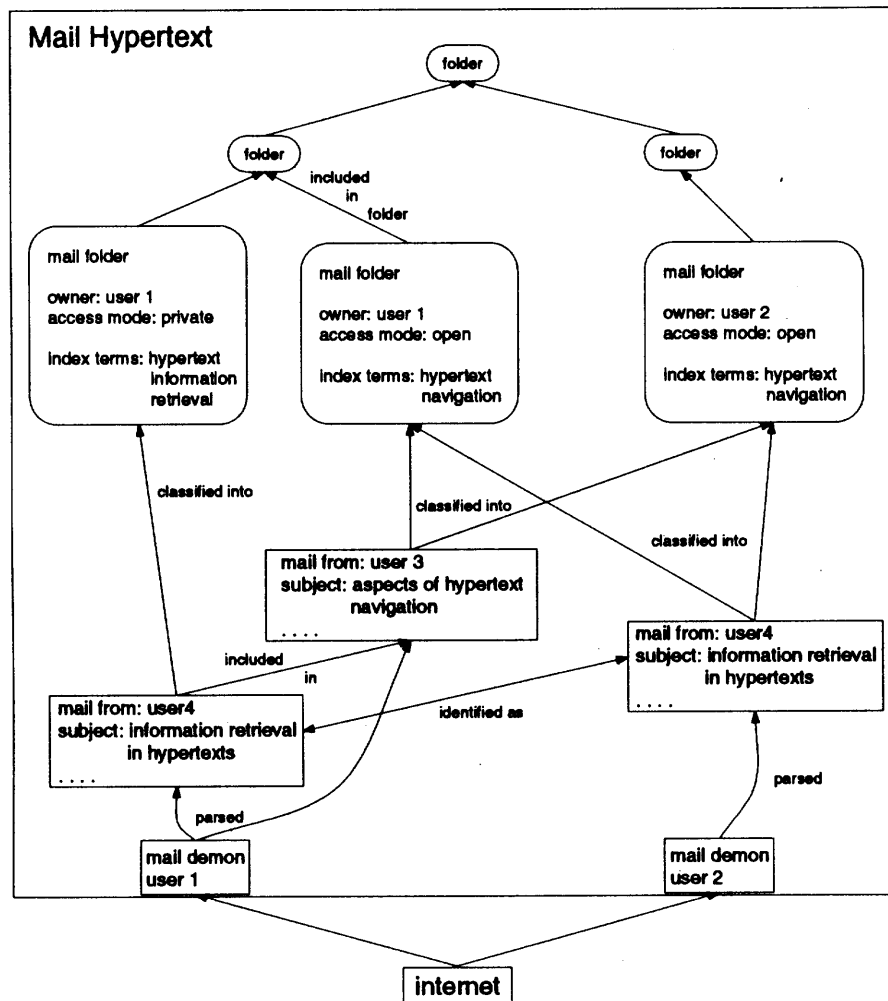


Fig. 1. 'User 2' receives mail which is classified into the two best fitting folders which 'user 2' has access to. 'User 1' receives the same mail as an included file from 'user 3'. The mail items are separated, the identity of one mail item already contained in the archive is detected, the units are connected by links and classified. The included mail is classified into the best fitting folder accessible to 'user 1' (but not to 'user 2').

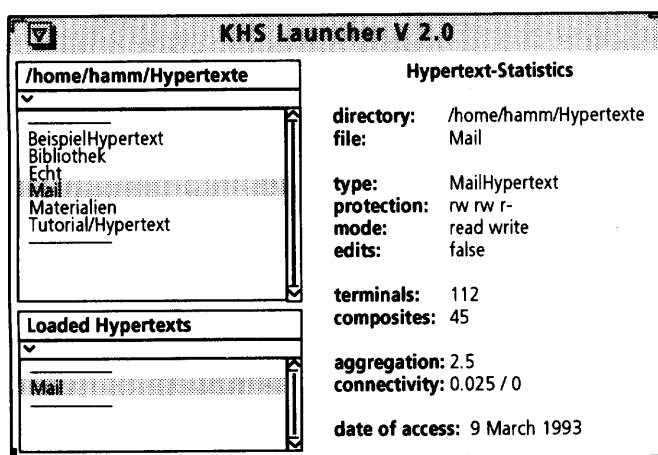


Fig. 2. The KHS launcher gives a list of available and a list of loaded hypertexts. Information about the selected hypertext is provided additionally.

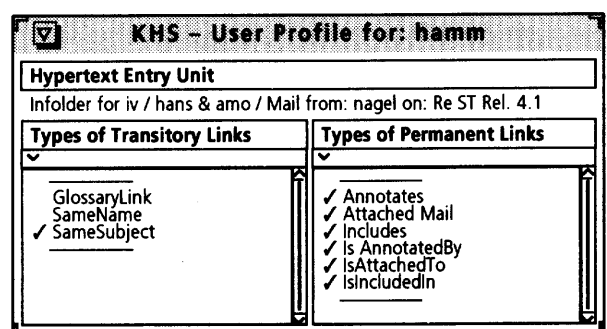


Fig. 3. This tool allows the user to set the unit that a hypertext session will start with, the types of transitory links which can be computed by the system and the types of permanent links which are actually contained in the hypertext. Only the selected types are used by other KHS tools.

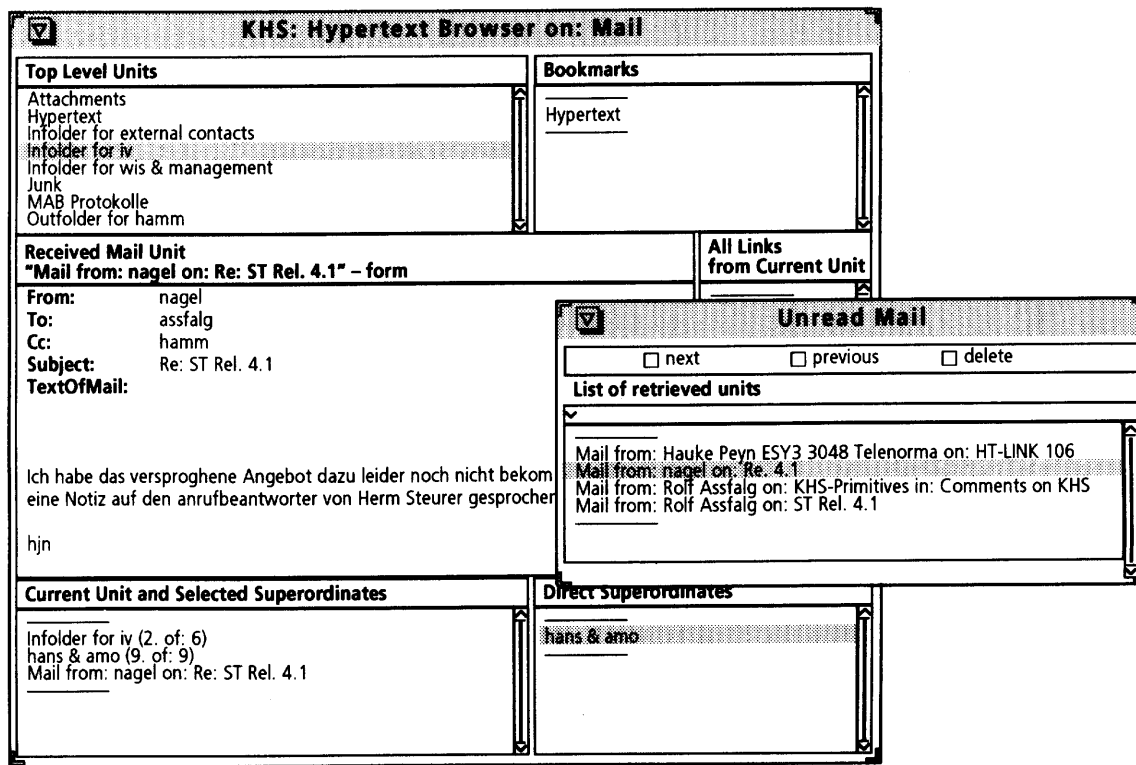


Fig. 4. The KHS browser shows a received mail chosen from a list of unread mail in its context.

hypertext, an active process will be started which looks for incoming mail throughout the hypertext session.

The browser is the most important interactive KHS tool (see Fig. 4). It is designed to show a single hypertext unit in its context, allowing the user to read the unit, to navigate within the structure hierarchy or the link web forming the hypertext, or to edit the content of a unit or its contextual embedding in the hypertext. Additional services may be started from here.

Readers of a mail hypertext will probably ask for unread mail first. KHS will answer by presenting a list of unit names. Selection from this list will lead to an update of the KHS browser so that the content of the selected units can be seen. If there are any included mails or files attached to the mail unit, the users can select the corresponding links from a list of typed links, and an additional window containing the file content will be opened.

Users may now wish to change the properties of one of the received mail units; for instance, they may

change the protection mode, describe it by index terms or set a date for it to be redisplayed and remind them of something still to be done. In this case, the users will start the unit feature browser which supports these tasks. If this does not fit the classification as proposed by the system, the unit can easily be moved to some other already known or newly-created folder.

Searching for archived mail is done by filling the slots of a prototypical mail unit. A set of all matching units will be displayed and can be browsed by the user.

7. Current and further issues in the development of KHS

The e-mail application, as described above, is operational and in use in the Information Science Department at the University of Constance. Some aspects of the further development of KHS are briefly reviewed in this Section.

7.1. Other sources of external information

There are two tracks in improving access to external information.

- (1) Information is retrieved and downloaded. The result is included into the hypertext automatically. This approach is appropriate to information which may be retrieved only once (e-mail) or at high cost. An interface to bibliographic online databases has already been implemented [14].
- (2) Special KHS units serve as gateways to external sources. When the user navigates to such a unit, a connection to the host will be established and the corresponding information retrieved. Gateway units behave in a similar way to other KHS units; therefore, readers do not need to know whether they are currently using internal or external information. This approach is useful when information can be retrieved at low cost or is changing rapidly. We will use this approach to connect KHS to value-added services, such as gopher.

7.2. Database support

KHS is implemented in Smalltalk-80 and the features described above are fully operational. KHS uses a file system-based method for the permanent storage of hypertexts. The advantage of this approach is that core functions of KHS (but not, for instance, for e-mail) are available on several platforms (Unix systems, PC using Windows, Apple McIntosh). The disadvantages are the low performance and the difficulties of coping with huge quantities of data. Therefore, an alternative storage model based on an object-oriented database (using GemStone) is currently being implemented.

8. Concluding remarks

Open systems and seamless integration of information streams are among the main issues in information science for the near future. Hypertext is one possible answer to the questions which arise in this context, but only hypertexts which are both controlled and free at the same time. Typed hypertext objects provide the integrating factor, giving special support for different applications within a uniform interaction environment. The example of electronic mail has shown how an elaborated application – supplying more functionality than the standard software – can be integrated into this framework easily. The combination of mail and other applications within the whole hypertext

system is then a minor problem. The openness of hypertext is thus no longer a restrictive barrier.

References

- [1] R. Afalg, R. Hammwöhner and M. Rittberger, Linking Internet capabilities within hypertext. In: *Submitted to the Seventeenth International Online Information Meeting, London* (1993).
- [2] R. Akscyn, D. McCracken and E. Yoder, KMS: distributed hypermedia system for managing knowledge in organizations. In: *Hypertext '87 Papers, University of North Carolina, Chapel Hill, NC* (1987) 1–20.
- [3] M. Bernstein, An apprentice that discovers hypertext links. In: *Hypertext: Concepts, Systems and Applications: Proceedings of the European Conference on Hypertext, INRIA, France* (1990) 212–223.
- [4] K.S. Catlin and N. Garrett, Hypermedia templates: an author's tool. In: *Proceedings of Hypertext '91, San Antonio* (1991) 147–160.
- [5] M.A. Casanova and L. Tucherman, The nested context model for hyperdocuments. In: *Proceedings of Hypertext '91, San Antonio* (1991) 193–201.
- [6] S.J. DeRose, Expanding the notion of links. In: *Proceedings of Hypertext '89, Pittsburg* (ACM, 1989) 249–258.
- [7] H. Davis, W. Hall, I. Heath and G. Hill, Towards an integrated information environment with open hypermedia systems. In: *Proceedings of ECHT '92, Milano* (1992) 181–200.
- [8] D.C. Englebart, Authorship provisions in Augment. In: *Proceedings of the IEEE COMPCON* (1984) 465–472.
- [9] E.A. Fox, From information retrieval to networked multimedia information access. In: *Information Retrieval '93. Von der Modellierung zur Anwendung*, edited by G. Knorz, J. Krause and C. Wormser-Hacker (1993) 116–124.
- [10] M.E. Frisse, Information retrieval from hypertext: update on the dynamic medical handbook project. In: *Proceedings of Hypertext '89, Pittsburg* (1989) 199–212.
- [11] R.J. Glushko, Design issues for multi-document hypertexts. In: *Proceedings of Hypertext '89, Pittsburg* (ACM, 1989) 51–60.
- [12] F. Garzotto and P. Paolini, HDM – a model for the design of hypertext applications. In: *Proceedings of Hypertext '91, San Antonio* (1991) 313–328.
- [13] F.G. Halasz, Reflections on NoteCards: seven issues for the next generation of hypermedia systems, *Communications of the ACM*, 31(7) (1988) 836–852.
- [14] R. Hammwöhner and M. Rittberger, KHS – ein offenes Hypertext-System. In: *Information Retrieval '93. Von der Modellierung zur Anwendung*, edited by G. Knorz, J. Krause and C. Wormser-Hacker (1993) 208–222.
- [15] M. Hofmann, U. Schreiweis and H. Langendörfer, An integrated approach of knowledge acquisition by the hypertext system CONCORDE. In: *Hypertext: Concepts,*

- Systems and Applications: Proceedings of the European Conference on Hypertext, INRIA, France* (1990) 166–179.
- [16] R. Kuhlen and M. Hess, Passagen Retrieval: – auch eine Möglichkeit der automatischen Verknüpfung in Hypertexten. In: *Information Retrieval '93. Von der Modellierung zur Anwendung*, edited by G. Knorz, J. Krause and C. Wormser-Hacker (1993) 100–115.
- [17] E. Lutz, H.v. Kleist-Retzow and K. Hoerning, MAFIA – an active mail-filter-agent for an intelligent document processing support, *ACM SIGOIS Bulletin* 11(4) (1990) 16–32.
- [18] T.W. Malone, K.R. Grant, K. Lai, R. Rao and D. Rosenblitt, Semi-structured messages are surprisingly useful for computer-supported coordination. In: *Computer Supported Cooperative Work Conference* (1987).
- [19] T.W. Malone, K.R. Grant, F. Turbak, S. Brobst and M. Cohen, Intelligent information-sharing systems, *Communications of the ACM* 30(5) (1987) 390–402.
- [20] A. Reinhardt, Smarter e-mail is coming, *Byte* (1993) 90–108.
- [21] G. Salton, J. Allan and C. Buckley, *Automatic structuring and retrieval of large text files* (Technical Report TR 92–1286) (Cornell University, Ithaca, USA, 1992).
- [22] R.H. Trigg and M. Weiser, TEXTNET: a network-based approach to text handling, *ACM Transactions on Office Information Systems* 4(1) (1986) 1–23.