Universität Konstanz
Informationswissenschaft
Postfach 5560
78434 Konstanz
e-mail: michels@inf-wiss.uni-konstanz.de

# A Homogenous Interaction Platform for Navigation and Search in and from Open Hypertext Systems

Marc Rittberger/Rainer Hammwöhner/
Rolf Aßfalg/Rainer Kuhlen

Bericht 41-94
(WITH-7/94)
Februar 1994

ISSN 0942-2625

# A Homogenous Interaction Platform for Navigation and Search in and from Open Hypertext Systems[1]

**Rittberger, M; Hammwöhner, R; Aßfalg, R.; Kuhlen, R.**

**Department of Information Science, University of Constance,**
**D-78434 Konstanz, Germany**
**e-mail:{ritt,hamm,assfalg,kuhlen}@inf-wiss.uni-konstanz.de**

## Abstract

**In response to the challenge of open hypertext systems allowing access to heterogenous information resources in the world-wide commercial and scientific information market, we propose a hypertext system model and architecture based on typed objects and links making possible semantically controlled access to information units within the hypertext, as well as to the outside information systems world. We concentrate particularly on a common interaction platform by describing the different browser types of the Constance hypertext system (KHS), and describe in more detail various different navigation techniques and term-based retrieval methods - thus furthering a productive partnership between information retrieval and hypertext.**

[1] This article is based on a paper submitted to the 17th Annual Conference on Research and Development in Information Retrieval (SIGIR94) to be held in Dublin, Ireland, from 3-6 July 1994

# 1 Introduction

Recent developments have tend to create more open hypertext systems with the positive effect that the information world is becoming increasingly open to flexible navigation and retrieval by local hypertext developers and users. By open hypertext systems we understand systems which are, among other things, open for various types of documents, open for various media objects, open for heterogenous information resources, external value-added services from Internet included, and open for users with various perspectives and information needs. Unfortunately, however, the openeness of hypertext systems has the somewhat negative consequence that the local owner or user of the hypertext has only a limited degree of control over incoming material. Given the heterogenous nature of resource data, this management problem is accompanied by an even increasing need for control and guidance. This may be called the open hypertext systems dilemma.

In the following we take for granted that a local hypertext system (such as KHS, which will be focused on in this paper) allows access to the basic and value-added services of Internet, thus providing not only continuous incoming information from e-mail services, list servers, bulletin boards, usenet news, but also particularly requested information from world-wide services of online catalogues (OPACs), menu-driven gophers, distributed data banks (WAIS) and other external hypertext systems following the WWW-protocol. In addition, it provides access via DATEX-P (or the Telnet interface) to the international commercial information market with its several thousand data bases, from which new information must occasionally be incorporated into the local hypertext base. This external information material is added to the equally heterogenous information produced by local research groups (internal e-mail, reports, course descriptions, protocols of meetings, etc.). All the diverse sorts of information must be manageable by systems such as the Constance hypertext system (KHS).

From a technical perspective, the cooperation of the different Internet value-added services can already be considered as forming a world-wide integrated information system, but from an information methodology point of view we find a highly inadequate integration of the heterogenous interaction styles (access, navigation and query formulation and search facilities) and, perhaps even worse, only very limited integration of different knowledge representation techniques (such as automatic classification, indexing, abstracting or even high-level knowledge-based content analysis). In this paper we concentrate on the problem of providing users with an integrated interface that allows generally homogenous interaction with heterogenous resources. We suggest solutions which go beyond the performance of client programs already on the market such as X-Mosaic or Cello.

At the beginning of hypertext development there was a somewhat controversial debate concerning whether techniques stemming from the information retrieval world (controlled vocabulary, Boolean-based query formulation, or advanced automatic techniques such as association factors, ranking algorithms, clustering procedures, etc.) should be embedded into the hypertext world, which has shown a preference for concepts such as creative associative browsing and explorative navigation. This debate has proved to be unproductive, in particular with respect to large real world hypertext bases and, naturally, in the context of open hypertext systems. These afford a fruitful combination of controlled search techniques and equally semantically controlled and guided navigation tools, which can be considered a special form of searching, namely explorative searching. Information retrieval and hypertext have never been hostile at odds, but rather productive partners [1, 2, 3, 4].

The KHS architecture and model which will be described in the second section below proposes a solution based on typed objects (units and links) to the challenge of controlling heterogenous materials. Section 3 describes in more detail some of the navigational tools which are available through KHS. Section 4 proposes new (and well known) solutions to the problem of content based searches (within the information retrieval paradigm) which are available with the KHS query browser. Similar to KHS browsers intended for more navigation-oriented searching, this browser provides a basis for uniform interaction with internal and external resources.

## 2 The hypertext-model of KHS

KHS is an open hypertext-system designed to allow the integration of various application domains, the use of multiple information resources and parallel use by an – in principle – arbitrary number of users. The unifying framework is supplied by a generic, application independent hypertext-model comprising a structure model which describes the structure of well-formed hypertexts with an interaction model which defines generic interaction styles. Both the structure model and the interaction model can be refined to suit the needs of special applications or individual users.

### 2.1 The structure model

The simple node-link structure of early hypertexts proved unable to provide sufficient orientation clues in large and complex hypertexts [5]. KHS therefore employs additional structuring mechanisms as follows:

1. Typing of hypertext objects [6] allows the stepwise refinement of the structure and behavior of hypertext objects.
2. Semi-structured hypertext objects offer structured data where they are required for further inference processes.
3. Composite nodes provide a polyhierarchical structuring mechanism.

#### *2.1.1 Typing of hypertext objects*

The type of a hypertext object determines its internal structure (content) and behaviour. KHS distinguishes between two fundamentally different object types: units and links.

Typing of units Hypertext units are devoted to the representation of the information content of a hypertext, whereas links realize the relations between such items of information. The 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 into composite units containing further units (see section 2.1.3) and media units containing text, images or sound combined with structured data (see section 2.1.2).

Typing of links Imposing appropriate links on a set of distinct discourse elements may transform the set into a cohesive interconnected discourse. The large number of possible connections which exist between two (or more) units (hypertext nodes and links) necessitates an explicit differentiation of link types based on to their function in the discourse.

Depending on their type, links may connect whole units, pieces of text (hotwords) within units, or sections within images (hotarea).

### *2.1.2 Semi-structured hypertext objects*

Semi-structured information-objects have already proved useful in the context of cooperative work [7], thus any type of KHS-object provides a structured part which may be used for retrieval purposes.

Each KHS object allows structured access to the following data:

1. who created (modified) the object and when did this happen,
2. is it 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 only temporarily (for the current session only).

Each hypertext-unit may additionally be described using a set of index-terms.

Subtypes of the so-called form units contain further application specific enhancements of their structured parts. For instance, hypertext units representing received mail allow access to the sender, recipients, subject, message-id., etc. of a specific mail item.

### *2.1.3 Composition of units*

The shortcomings of a mere node/link hypertext structure [5] have led to the concept of composite hypertext nodes which may themselves contain nodes[2]. The KHS hypertext model regards composite nodes as the backbone of the hypertext structure and as a means for structured navigation. Any unit within a hypertext – with the exception of a single, so-called "top level unit" – 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 presently selected unit within the structural hierarchy can be displayed for the user, thus providing orientation.
2. The subunits of a composite unit are ordered, thus providing a path mechanism which can be followed sequentially. A unit can be embedded into several paths, thus building forking or converging paths (see also [10]).
3. If a unit is multiply embedded into the multihierarchy of composite units, each of these embeddings may serve as a context of interpretation permitting context specific unit contents, or various sets of links to be displayed (see also [11, 12]).
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, they can anticipate the kind of information they will find.

[2] Early hypertext-systems which allowed composition are Textnet [8] and Augment [9]

## 2.2 The interaction model

### *2.2.1 KHS — a set of interacting tools*

KHS employs a multi-window interface to its hypertexts. The design of this interface is not guided by one of the well-known metaphors like *book*, *card stack* or *network*. The major design assumptions of the KHS interface are the following: At any given point in time reader's attention is concentrated on a particular hypertext unit. The unit's content and a minimal set of contextual information (embedding in the structural hierarchy, outgoing links, etc.) are displayed within one central tool, the Hypertext Browser (see section 3). Interaction with the unit takes place via a mouse click on a hotword or on lists of unit names. More complex functions can be activated by unit type-specific pop-up menus.

Nevertheless, no single tool can satisfy the presentation and interaction demands of a complex hypertext model. Therefore KHS provides a set of tools which can be additionally activated, providing access to:

1. special properties of hypertext objects (units or links),
2. the content of additional units,
3. lists of units obtained by search processes,
4. the dialogue history,
5. overviews of the structural hierarchy and link webs

All of these tools communicate with each other and thus guarantee a consistent display of the actual state of the hypertext. After clicking on a unit's representation within an overview, for instance, the unit's content/properties, etc. will be displayed within the other respectively active tools (see section 3).

### *2.2.2 Discourse clues during navigation*

Working with KHS will in any case include exploratory interaction styles. When navigating through a hypertext (or a relevant subset), a user must take decisions as to which unit he regards as the most appropriate one to be read next. KHS assists him by providing as many discourse clues as possible to indicate where the next navigation step will lead [13].

KHS has three styles of navigation:

1. Traversal of hierarchies: The user follows an author– or system-defined path. His reading attitude is the same as when reading a book. Therefore KHS provides only limited meta-information between the navigation steps.
2. Exploration of relevance sets: The user has performed a search within or beyond the boundaries of the hypertext. As a result, he obtains a set of relevant units which he may want to explore. In this case the user knows about the common features of the retrieved units. Display of a unit's name and type will give an adequate indication of its content and structure.
3. Link navigation: The most extreme degree of uncertainty with respect to the destination unit's content and function arises from link navigation, especially when initiated from hotwords. Therfore KHS provides more meta-information, including type of link, type and name of destination unit and, if available, iconic representations of images contained in the unit (see section 3).

#### *2.2.3 Filters*

Often, especially within highly interconnected hypertexts, too much information is retrieved by a single navigation or search step. Like many other hypertext-systems, KHS offers filter systems which prevent any information from being presented which does not conform to special filter conditions. The most important KHS filter types are type and structure oriented:

1. Type based filters preclude the presentation of any units or links which do not conform to one of a set of previously chosen types.
2. Structure based filters (see also [14]) only regard units (and links which lead to these units) as relevant which are embedded into special branches of the multihierarchy. As these structures may be constructed dynamically and temporarily (e.g. as result of a search), these filters can be used to combine the search results of several queries.

#### *2.2.4 Choosing the appropriate context*

Whenever a hypertext unit is reached during navigation, one of several possible embeddings within the multihierarchical structure must be chosen which will serve as the interpretive context (see section 2.1.3). KHS provides a set of context selection strategies which can be used in combination. The context of departure (the currently displayed context) defines which combination of strategies will actually be chosen.

The context selection rules determine the selection of the context:

1. with the least depth within the hierarchy,
2. which shares the most units with the presently displayed context,
3. which (depending on its index terms) best fits a user-defined interest profile (query),
4. which best fits a system-defined interest profile (user model),
5. which thematically best fits the units explored during the last navigational steps.

## 3 Navigation in KHS

In the previous sections the hypertext model of KHS was described. In the following it is demonstrated how these concepts are employed, how fundamental concepts such as link-types and unit-types are used as basic mechanisms for navigational support, and how they are applied using the unique interaction model of KHS.

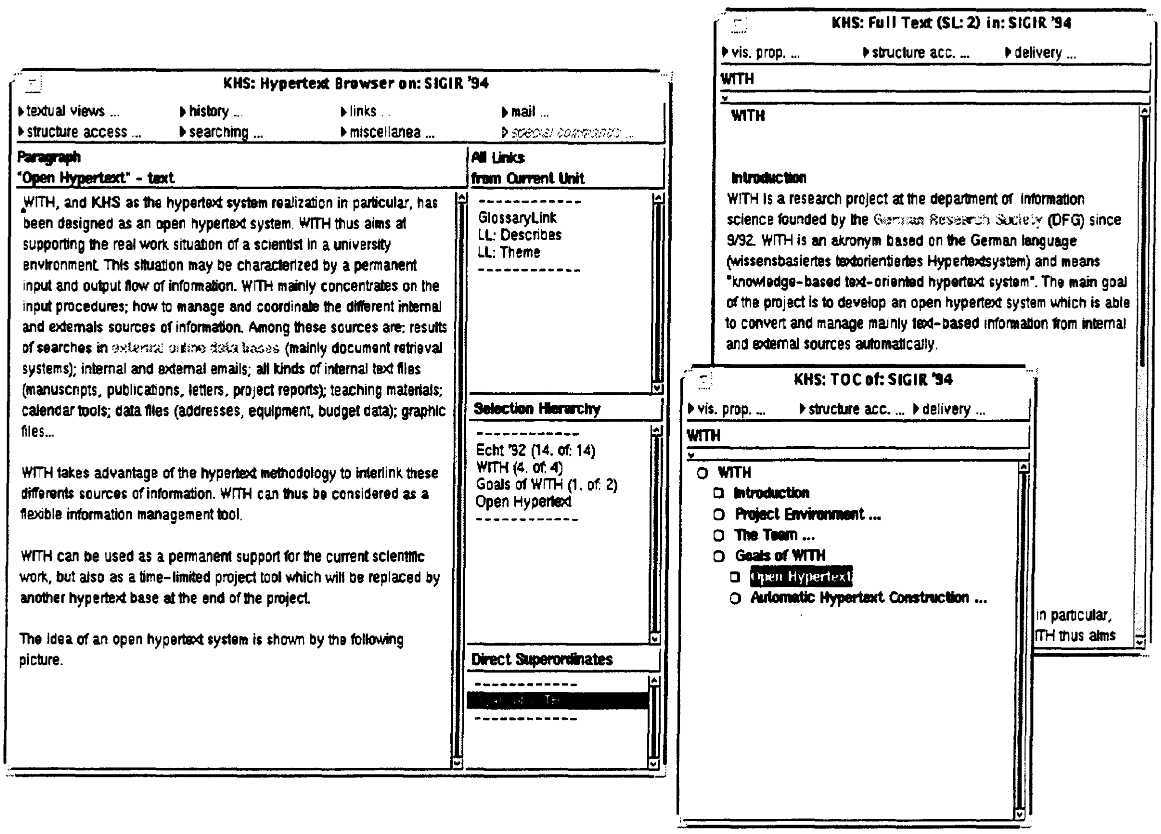
### 3.1 Navigation Tools

#### *3.1.1 The KHS Browser*

The main KHS working tool is the KHS-browser (see figure 1). The KHS browser is both an editing tool and a navigational tool displaying the current hypertext unit. The user can recognize how the currently selected unit is embedded in the hypertext's polyhierarchical structure, and he can recognize the links and their type.

Generally, a user of the KHS-browser can:

1. select a sub-unit from the table of contents of a composite unit,
2. select a unit from a list showing the contextual embedding of all units from the presently selected unit on up to the root of the hypertext's structural hierarchy,



WITH, and KHS as the hypertext system realization in particular, has been designed as an open hypertext system. WITH thus aims at supporting the real work situation of a scientist in a university environment. This situation may be characterized by a permanent input and output flow of information. WITH mainly concentrates on the input procedures; how to manage and coordinate the different internal and externals sources of information. Among these sources are: results of searches in external online data bases (mainly document retrieval systems); internal and external emails; all kinds of internal text files (manuscripts, publications, letters, project reports); teaching materials; calendar tools; data files (addresses, equipment, budget data); graphic files...

WITH takes advantage of the hypertext methodology to interlink these differents sources of information. WITH can thus be considered as a flexible information management tool.

WITH can be used as a permanent support for the current scientific work, but also as a time-limited project tool which will be replaced by another hypertext base at the end of the project.

The idea of an open hypertext system is shown by the following picture.

WITH

**Introduction**

WITH is a research project at the department of information science founded by the German Research Society (DFG) since 9/92. WITH is an akronym based on the German language (wissensbasiertes textorientiertes Hypertextsystem) and means "knowledge-based text-oriented hypertext system". The main goal of the project is to develop an open hypertext system which is able to convert and manage mainly text-based information from internal and external sources automatically.

in particular,
ITH thus aims

Fig. 1 The KHS Browser, Table of Contents Browser and the Full Text Browser

3. change the context of a presently selected unit. This is not a navigational step, but changes the contextual interpretation of the current unit, possibly also changing the unit's contents and available navigational options,
4. select a hotword from the text for the purpose of navigation.
5. navigate through successive selections of a link-type and a concrete link.

### *3.1.2 The Table of Contents Browser*

This tool (see figure 1) provides a view of the current unit's embedding into the hypertext's polyhierarchic structure. The user can navigate through this structure, selecting any item displayed in the table of contents browser. Filter-mechanisms allow users to reduce the amount information being displayed, so that the table of contents browser will show only the information surrounding the presently selected unit, but omit the sub-structures of other contexts. Several filter-mechanisms, in particular the fisheye-principle, are described in [15].

### *3.1.3 The Full Text Browser*

The Full Text Browser (see figure 1) allows linearized, scrolling-oriented access to the complete text of all units contained in any subhierarchy. The same filtering mechanisms as described in the previous section on the Table of Contents Browser are employed here as well. Futhermore, a user still can use navigation-facilities similar to those in the KHS browser he is already accustomed to.

## 3.2 Navigation in a Structure consisting of Composite Nodes

As introduced in section 2, all units of a KHS hypertext are arranged in a polyhierarchy which allows the definition of domains (composite units) devoted to special topics, applications, or user groups. Thus the hypertext model of KHS permits one and the same terminal node to be integrated into various composite units. If the user navigates downwards through the hypertext's hierarchy, a variety of terminal nodes can be reached by navigating along various paths. The different paths probably represent different topics or subjects. Thus the information in the hypertext is stored in only one location, but can be embedded in different contexts. In the following it is explained how this principle is practically applied:

The results of an online search performed by a user (detailed in [16, 17]) are automatically classified to several contexts such as:

1. the author context
2. the publication context
3. the publication year context
4. the document type context

Using this structure, the user can, for instance, navigate through the publications of a single author, or through all the articles in a particular book.

This approach is rather similar to the 'design of linking' in [18]. Another application of multiple contexts is to organize mail archives using KHS. A single piece of mail is integrated so that a user can navigate through composite units to reach a specific archived mail-document. Examples of what such composite units are standing for:

1. the sender of a mail
2. the receiver
3. the topic
4. a predefined keyword which is part of the mail's subject

If an incoming e-mail document is shared by several KHS users, there will be only one copy of the document, which will be integrated into all recipients' contexts.

This principle is useful, because it generates different views of a particular piece of information to be explored employing the KHS navigational tools. Consequently, the user is more likely to make an adequate association while navigating through the hypertext. The result is: Embedding a particular piece of information into multiple contexts generates 'value-added' effects.

## 3.3 Link Navigation

Conklin has shown that when a user encounters a link he must decide whether following a side path is worth the distraction. Conklin called this the problem of 'informational myopia'. Therefore Landow postulated a 'rhetoric of arrival', because the user cannot see what there lies behind a given link. Moreover: 'If the reader begins to fear that she is overlooking the crucial inormation or if she feels lost in a maze of hypertext links, the reader will abandon hypertext and insist upon conventional media.' One way of orienting the user is to indicate the semantics of the linkage by using typed links [19, 13, 20].

### *3.4.1 Internet Resources*

Connections among computers are called 'Internet connections' when they communicate using the TCP/IP-protocol. Based on advanced interaction protocols such as Gopher, WWW, and Z39.50 or simply on the concept of a "Network Virtual Terminal" (NVT), it is possible to interact with systems such as online databases, menu driven information systems, hypertext-like information systems, or online public access catalogues (OPAC's). (detailed in Ed Krol's Internet book: [21])

Gopher can be integrated into the data and interaction model of KHS by means of a special unit type (Gopher Unit). Units of that type do not represent media or textual information, but pointers to Gopher contents, specified by a Gopher address comprising the host name, the directory-path, and the Gopher-Document's type-ID. As soon as a user navigates through a hypertext to a KHS Gopher unit, KHS links up with the Gopher server in order to obtain data. This principle allows the user to employ the same navigational tools. Moreover, a user can in this way gain access to the most up-to-date information. With Gopher units, the user can click any item, and will then be guided to another piece of information located somewhere in the 'Gopher space'. Thus the user deals with a transparent access mechanism, because he interacts with a Gopher directory as well as with a KHS composite unit. The functionality of a Gopher-client program has thus been adapted to the uniform interaction model of KHS. Whenever a user works with a Gopher unit, he can use the following features:

1. Gopher-navigation by selecting items, similar to navigating with KHS composite units.
2. step-by-step Gopher-backtracking, similar to using the KHS history tool
3. possibility of ascending within the Gopher's menu structure, similar to ascending within the structure of a KHS hypertext.
4. navigation by direct entry of Gopher access information.

The user can browse through the Gopher-space from within KHS in order to explore or to integrate any suitable information into his current hypertext-context. A user who has navigated through the hypertext and reaching a 'Gopher-unit' can proceed further in the 'Gopher-space' by clicking on a KHS Gopher unit's text surface, in the same way as he would using a conventional Gopher client program. There are thus two levels of navigation, navigating on the KHS hypertext's units, and navigating through Gopher-space. Consequently, the user has access to Gopher units as well as to all other KHS hypertext units, because the same tools and principles are employed. With KHS one is able to conveniently organize all the information originating from these Internet services in his own personal way. This is an innovation compared with simple bookmark- and annotation-capabilities used in several multi-format viewers such as X-Mosaic or WWW standard browsers.

### *3.4.2 Default Queries*

When navigating towards a special type of composite unit, a query is made which is much more complex than a simple standard Gopher request of the sort mentioned above. Such a query could be directed, for instance, to online databases or to other external information resources. Hereby the user would have transparent access to information matching the query associated with this special type of composite unit. This function is

In KHS the user deals with different types of links. The KHS browser thus supports the user by showing the type of each link. There are several link-types, such as for example:

1. application specific links (e.g. a reply-link between two e-mail documents or a zoom-link to other images)
2. application independent links (e.g.: a glossary-link or a statistical link)

If he knows the link types the user can usually foresee what he will find when navigating in a specific direction. So the typing of links is an improvement, which decreases the effects of 'informational myopia'. In the following we describe how the typing of links with the KHS system can be of use when interacting with the system:

Suppose that a user reads the text of a unit. He recognizes hotwords within the text displayed in an emphasized mode. As he moves the cursor to a hotword, it changes shape, indicating whether there are one or more hotwords hidden in the emphasized section of text. If the user clicks the emphasized string, the system will, where needed, perform hotword disambiguation. Then a list of link types will be displayed which are connected to the chosen hotword. The user chooses a type with respect to its discourse function. A list of units will appear which can be reached by a link of the chosen type. This list will show a unit's type and name. If the destination unit contains an image, an iconic representation of the image will be shown. If a destination unit is chosen, navigation will be initiated. Depending on the navigational semantics of the respective link type, the information of the destination unit will be displayed alternatively or in addition to the presently displayed unit.

In the previous section the multiple context feature of KHS was described. Typed links are also useful for navigating through e-mail archives and online document records. During the process of integrating online document-records, KHS automatically generates typed links leading to the composite units representing the co-authors. Another example of this technique is linking e-mail. E-mail which has been received can be forwarded using the KHS e-mail browser. In this way the KHS user can generate mail containing other mail. To express this relation of inclusion, KHS offers a special link-type (the "includes"-link). KHS automatically creates a link of this type whenever the user is forwarding already existing mail. The advantage is that a particular e-mail is stored only once.

Besides the links used in textual units, KHS also contains image-links. A hypertext author can define rectangular areas in images linked to any given unit. If one image shows an enlarged detail of another, a hypertext author can define a 'zoom'-link indicating that the user, by selecting this link, can obtain the 'close-up' image.

### 3.4 Navigation in External Resources

In the previous sections we described how the KHS user can navigate within the system's resources or through results which have been previously generated by an online search. In the following we describe how users can gain access to external information employing both the same KHS navigational tools and the fundamental KHS interaction principles already described above.

### *3.4.1 Internet Resources*

Connections among computers are called 'Internet connections' when they communicate using the TCP/IP-protocol. Based on advanced interaction protocols such as Gopher, WWW, and Z39.50 or simply on the concept of a "Network Virtual Terminal" (NVT), it is possible to interact with systems such as online databases, menu driven information systems, hypertext-like information systems, or online public access catalogues (OPAC's). (detailed in Ed Krol's Internet book: [21])

Gopher can be integrated into the data and interaction model of KHS by means of a special unit type (Gopher Unit). Units of that type do not represent media or textual information, but pointers to Gopher contents, specified by a Gopher address comprising the host name, the directory-path, and the Gopher-Document's type-ID. As soon as a user navigates through a hypertext to a KHS Gopher unit, KHS links up with the Gopher server in order to obtain data. This principle allows the user to employ the same navigational tools. Moreover, a user can in this way gain access to the most up-to-date information. With Gopher units, the user can click any item, and will then be guided to another piece of information located somewhere in the 'Gopher space'. Thus the user deals with a transparent access mechanism, because he interacts with a Gopher directory as well as with a KHS composite unit. The functionality of a Gopher-client program has thus been adapted to the uniform interaction model of KHS. Whenever a user works with a Gopher unit, he can use the following features:

1. Gopher-navigation by selecting items, similar to navigating with KHS composite units.
2. step-by-step Gopher-backtracking, similar to using the KHS history tool
3. possibility of ascending within the Gopher's menu structure, similar to ascending within the structure of a KHS hypertext.
4. navigation by direct entry of Gopher access information.

The user can browse through the Gopher-space from within KHS in order to explore or to integrate any suitable information into his current hypertext-context. A user who has navigated through the hypertext and reaching a 'Gopher-unit' can proceed further in the 'Gopher-space' by clicking on a KHS Gopher unit's text surface, in the same way as he would using a conventional Gopher client program. There are thus two levels of navigation, navigating on the KHS hypertext's units, and navigating through Gopher-space. Consequently, the user has access to Gopher units as well as to all other KHS hypertext units, because the same tools and principles are employed. With KHS one is able to conveniently organize all the information originating from these Internet services in his own personal way. This is an innovation compared with simple bookmark- and annotation-capabilities used in several multi-format viewers such as X-Mosaic or WWW standard browsers.

### *3.4.2 Default Queries*

When navigating towards a special type of composite unit, a query is made which is much more complex than a simple standard Gopher request of the sort mentioned above. Such a query could be directed, for instance, to online databases or to other external information resources. Hereby the user would have transparent access to information matching the query associated with this special type of composite unit. This function is useful for questions which might be frequently posed, for example: "Which publications contain texts by author X", or: "How many patents exist for subject Y". Hypertext units containing such predefined questions either derive from the KHS composite unit (first sample question), or from a KHS terminal unit (second sample question). So they either temporarily contain the search result in the form of several terminal units, or they simply present the overall result in the own unit's text area.

The advantage of these default query units is that the user can navigate without being aware that he is making queries.

## 4 Searching and Retrieving in KHS

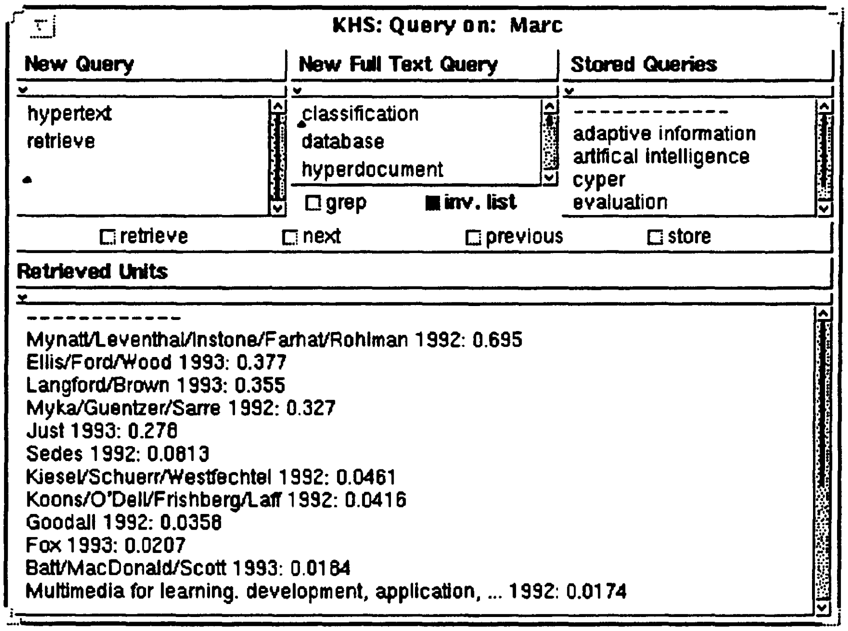
Besides explorative searching by browsing in an open hypertext environment, an information retrieval tool should also be available which gives the user direct access to any unit in a hypertext. In KHS we have developed a mechanism to retrieve precisely those units a user is searching for (e.g. a specific address) or to establish starting points in the hypertext for a content based navigational exploration. The need for finding such starting points is obvious, given that a normal hypertext collection may have to contain thousands of units in order to perform a successful navigational exploration in a hypertext environment [5, 22, 23].

In this section we focus on content based search and describe how the KHS query browser supports the search for various information sources within KHS by means of a single tool which can be used in a consistent manner throughout the search. We will first describe how the query browser works, then discuss how the user can access and integrate external information sources, and finally show how KHS allows us to reduce the information space needed for searching with the query browser.

### 4.1 Query Browser

The query browser display consist on four windows (see figure 2). Queries are formulated in the top three windows, and the lower window shows results as ranked by the system. In the upper left window one can search with index terms. Every unit in the hypertext has index terms, and most of the different unit types are automatically indexed by means of statistical or formal techniques. Terminal units with textual information use weighted index terms, which are calculated automatically with the help of term frequency and inverse document frequency [24]. When retrieving with the support of the query browser, the user can select index terms from a list. The resulting units can be arranged in alphabetical order, or on the basis of weights assigned to the index terms of the retrieved units. In the middle upper window the user has access to the full-text of all textual units of the hypertext. We use an inverted file which contains all stems of the words used in the hypertext. The user has several possibilities to enter terms for a full-text search:

1. The user can select index terms of the unit he is currently reviewing from the full-text query menu.
2. The user can select a group of one or more words from the text of the current unit and use the selected group of words as a single term for the search. A search using terms consisting of several words is possible, because the inverted file contains the position of every word in its unit.



hypertext
retrieve

classification
database
hyperdocument

------------
adaptive information
artifical intelligence
cyper
evaluation

```
------------
Mynatt/Leventhal/Instone/Farhat/Rohlman 1992: 0.695
Ellis/Ford/Wood 1993: 0.377
Langford/Brown 1993: 0.355
Myka/Guentzer/Sarre 1992: 0.327
Just 1993: 0.278
Sedes 1992: 0.0813
Kiesel/Schuerr/Westfechtel 1992: 0.0461
Koons/O'Dell/Frishberg/Laff 1992: 0.0416
Goodall 1992: 0.0358
Fox 1993: 0.0207
Batt/MacDonald/Scott 1993: 0.0184
Multimedia for learning. development, application, ... 1992: 0.0174
```

Fig. 2 The Query Browser with the search windows in the upper left and upper middle. The upper right window allows storing queries, and the bottom window presents the results in ranked order.

3. With the selection of a sentence or any textual part of an unit, every word in the selected part will be parsed and reduced to standard form. Every word can be used for a search in the full-text.
4. The user can choose terms from a list of words which contains in standard form all words used in the hypertext.
5. It is also possible to enter words by typing them into the middle upper window of the query browser.

The retrieved units are ranked as with searches with index terms. The user can also combine searches with index and full-text terms by using the upper left and upper middle windows together. The results are also presented in the bottom window.

The ranking of the results in the bottom window of figure 2 is done by summarizing the weights of the terms found by the system. The exclusive use of term weights for ranking would not provide the specific advantages of a hypertext. With the inclusion of links during ranking, there is an increase in the chances of finding good starting points in a network environment like hypertext [1, 2]. In our case a unit weight enlarges when two conditions are fulfilled:

1. The linked unit has also been found by the system for the current query.
2. The type of link between units is content based, like 'describe', 'explain' or 'define'. Links like 'has address' are not used.

The weight of the original unit is derived from the weight of all found and linked units ($W_{unit}$) multiplied by the weight of the link ($W_{link}$) and a reduction factor ($RF$). The new weight is calculated using the formula:

$$W_{new} = W_{old} + RF \cdot \sum_{RelUnits} (W_{Unit} \cdot W_{Link})$$

With the help of this method we can find units which are not only relevant in themselves, but are also connected to other query-relevant units. Finally, units will receive the highest weighting if they belong to a query-relevant cluster.

### 4.2 Accessing external information resources.

Often a user's need can be met simply by searching in locally available information sources. But sometimes there is no relevant, or not enough relevant, information available in the current information source. If this happens in KHS, the user can access external information sources like e-mail based services, online databases or gopher (see section 3.2). In the following we will describe the interaction necessary to access external commercially available online databases. Our main goal is to be able to use the same interaction model as that of the query browser for searching in a local hypertext.

Accordingly with our query browser, when searching in an online database the user only has to tell the system that he wishes to access not only internal, but also external information. The system will then search in an external database, instead of the locally available inverted file, and will present the retrieved documents in the same window as with other searches. With this window the user can access all the new semi–structured units and interact with these units as usual in KHS (e.g. delete them if they are not relevant).

Most current online databases work with Boolean logic and not the weighted searching familiar to the user of KHS from interaction with the query browser. Therefore a weighted search must be transformed into a Boolean one, or the weighted search must be run on the online host. In [25] and [26] experiments are described in which a weighted query was used in a local system and a search run on an online host (namely Datastar) using this weighted query formulation. An evaluation by [26] produced encouraging results finding that the results were equal or even better than those produced by a comparative Boolean search. We used the idea mentioned in [26] as an alternative to the internal search. A user can start an online search without having to alter the question formulated for his internal search. We use databases from the host STN-International, because all the necessary information needed to compute weights is accessible with the STN-International command language. The weight of the terms sent to the host is set to 1, and the weight of the answers is computed with:

$$Weight = \frac{tf(t_i, d_j)}{1 + tf(t_i, d_j)} \cdot idf(t_i)$$

where $tf(t_i, d_j)$ is the term frequency of a term $t_i$ in a document $d_j$. $idf(t_i) = \log \frac{N}{df(t_i)}$ is the inverse document frequency. $N$ is the number of documents in the database, $df(t_i)$ is the number of documents containing the term $t_i$. Several commands of the STN-International command language are necessary to compute the weights. The total number of Documents in the database is known, and the number of documents containing a term is available with the 'EXPAND' command, which shows part of the inverted file of the online database with the number of documents where a term occurs. The term frequency can be found with the help of 'DISPLAY OCC'. This command displays, for every document found in the online database, the number of times each search term occurs in each document. With this information, KHS computes the weights and selects the most relevant documents with the highest weights in the online database. The results are printed and sent to KHS. At KHS they are included in the polyhierarchical structure (see section 3), and a list of retrieved units is displayed in the bottom window of the query browser.

### 4.3 Defining and searching a context

The terms used in a specific search can be stored for later use in the upper right window. The user must give each stored search a name. In a later KHS session he can then access the stored search terms using the assigned names.

Besides search terms, it is also possible to store the results of a search with the help of the query browser. Since these are stored in a special context (see section 2), a new composite unit will be constructed by the system and all the units which have been retrieved in a index –, full-text –, online – or combined search will be stored below this new composite unit. If necessary, the user can build a hierarchical structure using the retrieved units. Such a hierarchy can be created in the new composite unit with the help of the index terms. Units having similar index terms are combined in new composite units named with the common index terms. The new hierarchy, with all the retrieved units, will still be located in the new context below the composite unit which was previously created.

Often users already know the area they must look for particular information. In such cases, KHS allows them to reduce the information space which must be searched (see section 2.2.3). To limit the search area they must first navigate to the highest relevant unit of the desired search context. By selecting this unit as the relevant context, they can then confine their search to the chosen unit and all subunits in the hierarchy below the chosen unit.

## 5 Concluding remarks

Although KHS provides a generally homogenous interaction interface to heterogenous resources on the basis of typed, semantically controlled or semi-structured objects and links, we are quite aware that we cannot and indeed do not wish to homogenize the broad variety of information systems and services found in the international scientific and commercial information market. This variety contributes to the creative potential of the information world which, nevertheless, should be easily accessed, comfortably searched, and creatively navigated, both in the local hypertext and the outside information world. KHS provides a common interaction platform for these purposes.

## Bibliography

1. M. Frisse and S. Cousins, "Information retrieval from hypertext: Update on the dynamic medical handbook project," in *Proceedings of the Hypertext '89 conference. (Pittsburgh, PA, 5.-8. Nov.)*, pp. 199–212, New York: ACM, 1989.

2. W. Croft and H. Turtle, "Retrieval strategies for hypertext," *Information Processing & Management*, vol. 29, no. 3, pp. 313–324, 1993.

3. G. Salton and J. Allan, "Selective text utilization and text traversal," in *Proceedings Hypertext'93*, pp. 131–144, 1993.

4. R. Kuhlen, *Hypertext und Information Retrieval — mehr als Browsing und Suche*, vol. R. Kuhlen (ed.): Experimentelles und praktisches Information Retrieval. Festschrift für Gerhard Lustig, pp. 309–324. Universitäts-Verlag Konstanz: Konstanz, 1992.

5. F. Halasz, "Reflections on notecards: Seven issues for the next generation of hypermedia systems," *Communications of the ACM*, vol. 31, no. 7, pp. 836–852, 1988.

6. R. Hammwöhner and R. Kuhlen, "Semantic control to open hypertext systems by typed objects," Tech. Rep. 34-93 (WITH-6/93), Universität Konstanz, Informationswissenschaft, 1993.

7. T. Malone, K. 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.

8. R. Trigg and M. Weiser, "Textnet: A network-based approach to text handling," *ACM Transactions on Office Information Systems*, vol. 4, no. 1, pp. 1–23, 1986.

9. D. Englebart, "Authorship provisions in augment," in *Proceedings of the IEEE COMPCON*, pp. 465–472, 1984.

10. P. Zellweger, "Scripted documents: A hypermedia path mechanism," in *Proc. Hypertext '89, Pittsburg*, (New York), pp. 1–14, ACM, 1989.

11. M. Casanova and L. Tucherman, "The nested context model for hyperdocuments," in *Proc. Hypertext' 91, San Antonio*, pp. 193–201, 1991.

12. N. Delisle and M. Schwartz, "Contexts: a partitioning concept for hypertexts," in *Computer Supported Cooperative Work Conference*, 1987.

13. G. Landow, "Relationally encoded links and the rhetoric of hypertext," in *Proceedings of the Hypertext '87 conference. (Chapel Hill, North Carolina 13.-15. Nov.)*, pp. 331–343, New York: ACM, 1987.

14. J. Coombs, *Hypertext, full text, and automatic linking*, vol. J.L. Vidick(ed.): Proceedings of the 13th International Conference on Research and Development in Information Retrieval, pp. 83–98. ACM: New York, 1989.

15. R. Aßfalg and R. Hammwöhner, "Eine Navigationshilfe nach dem fish–eye–Prinzinp für das Konstanzer Hypertext System (KHS )," in *Mensch und Maschine – Informationelle Schnittstellen der Kommunikation. Proceedings des 3. Internationalen Symposiums für Informationswissenschaft (ISI '92)* (H. Zimmermann, H.-D. Luckhardt, and A. Schulz, eds.), no. 7 in Schriften zur Informationswissenschaft, pp. 287–304, Universitätsverlag Konstanz, 1992.

16. R. Hammwöhner and M. Rittberger, "KHS – ein offenes Hypertext–System," in *Information Retrieval '93. Von der Modellierung zur Anwendung* (G. Knorz, J. Krause, and C. Womser-Hacker, eds.), no. 12 in Schriften zur Informationswissenschaft, pp. 208 – 222, Universitätsverlag Konstanz, 1993. Proceedings der 1. Tagung Information Retrieval'93.

17. R. Aßfalg, R. Hammwöhner, and M. Rittberger, "The hypertext internet connection: E–mail, online search, gopher," in *Online Information 93. 17th International Online Information Meeting, 7–9 December, London* (D. Raitt and B. Jeapes, eds.), pp. 453–464, Learned Information Ltd, 1993.

18. R. Kuhlen, "Aktivierung von Online-Informationsbanken aus Hypertextbasen," in *Online '91. 14. Europäische Congressmesse für Technische Kommunikation. Congress IV. Bürokommunikation: Konzepte und Strategien zur Unterstützung der Büroarbeit* (S. Sorg, ed.), pp. IV.01.01–IV.01.14, 1991.

19. J. Conklin, "Hypertext: An introduction and survey," *IEEE Computer*, vol. 20, no. 9, pp. 17–41, 1987.

20. M. Bernstein, "The bookmark and the compass: Orientation tools for hypertext users," *SIGOIS Bulletin*, vol. 9, no. 4, pp. 34–45, 1988.

21. E. Krol, *The whole Internet. User's guide and catalog*. O'Reilly and Ass. Inc:: Sebastopol, CA, 1992.

22. J. Savoy, "Effectiveness of information retrieval systems used in a hypertext environment," *hypermedia*, vol. 5, no. 1, pp. 23–46, 1993.

23. W. Janko, A. Taudes, and W. Faber, "Optimale startpunkte zur navigation in hypermediasystemen – ein entscheidungstheoretischer ansatz," in *Hypertext/Hypermedia '91, Tagung der GI, SI und OCG, Graz*, pp. 145–155, Mai 1991.

24. G. Salton and M. McGill, *Introduction to modern information retrieval. [dtsch.: Information Retrieval — Grundlegendes für Informationswissenschaftler.]*. Mac Graw-Hill : Hamburg/New York, 1983; 1987.

25. S. Robertson and C. Thompson, "Weighted searching: The Cirt experiment," in *Prospects for intelligent retrieval. Proceedings Informatics 10, King's College, Cambridge, 21–23 March 1989*, pp. 153–165, ASLIB, 1990.

26. H. Frei and Y. Qiu, "Effectiveness of weighted searching in an operational IR environment," in *Information Retrieval '93. Von der Modellierung zur Anwendung.* (G. Knorz, J. Krause, and C. Womser-Hacker, eds.), no. 12 in Schriften zur Informationswissenschaft, pp. 41–54, Universitätsverlag Konstanz, 1993. Proceedings der 1. Tagung Information Retrieval 93.