

# The Hypertext Internet Connection: E-mail, Online Search, Gopher<sup>1</sup>

Rolf Aßfalg, Rainer Hammwöhner, Marc Rittberger

Department of Information Science

University of Constance

## Preface

This article is based on a paper submitted to the 1993 "Seventeenth International Online Information Meeting". We would like to thank Rainer Kuhlen, University of Constance, and Dagobert Soergel, University of Maryland, for their valuable comments on this paper and the helpful discussions about open hypertext systems.

Constance, July 1993.

Rolf Aßfalg, Marc Rittberger, Rainer Hammwöhner

## Abstract

In this paper we show how to handle and organize the large amount of information accessible through the Internet or other public communication networks in a hypertext environment. The C(K)onstance-Hypertext-System (KHS) uses typed units to indicate the differences and the content and structure of information, comprising text, forms, images pointers to external information. We show how to imbed Internet services, which usual require rather different interaction styles, such as point-to-point communication (e-mail query formulation (online databases) or browsing (Gopher) into the uniform interaction model of the KHS. The integration of Internet services in an open hypertext environment produces value-adding effects which are also discussed.

## 1. Introduction

### 1.1 Internet fundamentals

In 1969 the Internet was born as a U.S. Department of Defense (DoD) experimental network called the ARPA-net, but nowadays the Internet consists of a collection of mail subnets administered by different organisations. Connections between computers are called 'Internet connections', if they are communicating using the TCP/IP-protocol. The Internet Protocol (IP) takes care of addressing and routing the data packets. These behave quite similar to an envelope which is transferred by the post office; the IP-protocol determines the address one has to use. The Transfer Control Protocol (TCP) is a transport protocol providing connection-oriented, end-to-end reliable byte data transmission. That is similar to a series of letters and the TCP-protocol watches for the correct order the sender has determined.[Kro92]

---

<sup>1</sup> This paper has been published in: D. Raitt and B. Jeapes (eds.), *Online Information 93. 17th International Online Information Meeting, 7-9. December, London*, pp. 453-464. Learned Information Ltd, 1993.



This text is published under the following Creative Commons Licence: Attribution-NonCommercial-NoDerivs 2.0 Germany (<http://creativecommons.org/licenses/by-nc-nd/2.0/de/>).

Based on the TCP/IP protocol some standard services such as:

- the File Transfer Protocol (FTP);
- the Simple Mail Transfer Protocol(SMTP);
- the Domain Name Service (DNS);
- the NetBios service;
- the TELNET Protocol.

have been implemented. In addition, supplementary services have been implemented such as:

- WWW (World Wide Web) protocol;
- the Gopher protocol;
- the CSO phonebook service;
- the WAIS (Wide Area Information Server) protocol;
- the NetNews service
- archie service

Furthermore, based on the TELNET protocol's concept of a "Network Virtual Terminal" (NVT), it is possible to interact with systems like online databases in a way quite similar to a simple terminal connection.

Some of these services mentioned above are used to build information systems where the users either communicate with one another or retrieve information from a huge volume of data. But these capabilities suffer from the numerous different interaction modes the user has to manage. Therefore, we propose the integration of such Internet services into an open hypertext system where special hypertext nodes carry out searches and the information retrieved is inserted into the hypertext and can thus be administered in uniform manner. We have integrated the following Internet services into our hypertext system:

- Electronic-mail;
- searching online databases using TELNET's NVT;
- functions of a Gopher client.

Before discussing the details of the integration of these Internet services, we will describe the KHS hypertext model.

## **1.2 KHS, an open hypertext system**

The opening of hypertext to an arbitrary number of users, to a variety of applications, and to the use of external information sources has a major impact to the use and administration of hypertext. The following problems arise:

- The quantity of information aggravates the disorientation problem already known from conventional hypertexts.
- A standardized interaction strategy of browsing and exploration will not suffice for all of the applications which should be supported by the hypertext.
- Some of the applications need access to structured data (the address of a person etc).
- New applications should be integrated efficiently.

- External information sources will not present their data in just the way the hypertext applications expect. Therefore transformation processes must take place which are specific to the type and source of information.

KHS addresses these problems as follows:

- All units of a KHS hypertext are arranged into a poly-hierarchy of composite nodes. According to this structure, KHS allows for the definition of domains, which are devoted to special topics, applications, or user groups. At any time of the hypertext navigation KHS informs the user which domain of the hypertext he is currently visiting.
- The type of a KHS unit defines the structure of its contents and its presentation- and interaction mechanism. Each unit type has its own adaptation to the general interaction model of KHS. Special type-specific transformation processes are devoted to the integration of external information into the hypertext; specifically, the Internet-based KHS services operate their own specific parsers. A special type of units - called form units - includes data fields in a record structure that allows access to properties of an object. Bibliographic units for instance include fields for title, author, publication year etc. Form units provide special interaction techniques as well, such as form filling and form matching.
- The type of a KHS link allows inferences on properties of the linked units. Therefore, link types may be used as a basic mechanism for navigation planning. KHS links may connect entire hypertext units or some partial information contained in the units, e.g. a piece of text or a single word (anchored links).
- A locking mechanism operating on hypertext units allows multi user access and the management of asynchronous events like the arrival of e-mail.

We will now discuss the integration of Internet services in the open hypertext system KHS.

## **2 Management of e-mail with KHS**

### **2.1 Properties**

E-mail has become an extensively used medium of information transfer and exchange which is well supported by modern user interfaces like Open Windows. The mail-intrinsic restriction to ASCII text is no longer visible to the user because adequate conversion routines are activated automatically. Nevertheless, some of the assumptions which determined the implementation of mail are not yet questioned:

- E-mail is still tailored to single users rather than work groups as receivers of mail.
- Archiving of mail is based on collections of flat files without interlinking or even cross search capabilities.
- Mail folder 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.
- Mail is treated as if it were only for information purpose. However, mail often contains some kind of to do lists. It should be possible to assign a date to a mail unit at which it will pop up again to remind one what still has to be done. Presumably there will be broad consensus that improved control over the handling of mail would be welcomed (and perhaps someone is already implementing it for Open Windows). The sharing of mail is not that obviously useful and therefore requires further motivation.

Our scenario in developing KHS is a group of scientists who manage their publications, memos etc. by means of hypertext. These scientists working in the same department and on related matters will very often get the same mail messages from external partners or mailboxes. Other items of mail are forwarded to the colleagues and of course some others are kept private. Taking this situation into account a shared mail archive seems to be desirable.

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

- Arriving mail is read by the system and converted into hypertext units automatically.
- The structural items of mail (author, subject) are accessible as properties of this unit.
- Every mail unit is classified automatically according to these properties. The classification leads to the insertion of the unit into one or more mail folder, which may be shared by several users.
- The privacy of mail if requested must be guaranteed.

## **2.2 Sorting mail for presentation and archiving**

The heart of an KHS mail hypertext is a system of mail folders used both for ordered presentation and for archiving of mail. Mail folders are implemented as special composite units organized in a hierarchy. Generic mail folders contain subfolders, whereas special mail folders are dedicated to sent and received mail. Index terms which are assigned to the folders describe the content of mail which is to be included. Every 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. Users may override the keep-time for individual units. After the set time, the mail item will be presented to the user for confirmation of the deletion. The shortest possible interval is one day, the longest one leads to permanent storage. This feature allows one to distinguish between mail items relevant only in the short run and others which should be kept for a longer time.

Mailboxes are special kinds of mail folders. Whenever arriving mail is included into a mailbox, it is distributed to its subscribers. Automatic mailboxes send on mail without any intervention, whereas moderated mailboxes send only mail approved by the moderator. Other mail-folders are able to initiate changes inside the archive, so that for instance a user is added to or deleted from a subscription list.

## **2.3 Processing of arriving mail**

Whenever the mail hypertext is active, an active process is watching for arriving mail. New mail will be read and parsed into a form unit filling the units slots with the mails author, message-id, recipients, subject, text etc. Based on the message-id the system checks whether the mail is already included in the archive. Mail items from lists, forwarded mail, and included mail often contain mail items which are not new to the user. If a mail archive is shared by several users, such duplicates will occur more frequently. If the mail is not found in the archive, the words of the mails subject and the name of its sender are used as index terms which are matched against the terms assigned to those mail folders to which the user is allowed to write. The mail item then is included in the mail folder with the highest matching score (several folders if there is a tie). 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 mail is guaranteed. Every user owns a mail folder readable only by himself for any

mail that does not match some other folder. Included mail and attached files receive special treatment. Included mail is cut from the mails text, parsed independently into a separate mail unit, and connected to the mail unit it originates from by an "included mail" link. The processing of attached files requires checking the type of the included data, decoding, and decompression so that a media unit representing these data can be created. At the time present KHS accepts text, bitmaps, PostScript graphics, and files as used by the Island-Tools as attached files.

## **2.4 Sending mail**

Sending mail requires filling a form with the mails recipient, subject etc. The recipient can be defined by a valid Internet address or by entering an alias. The resolving of aliases is achieved not 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, phone and fax-nr etc. This can be used not only by e-mail applications but can support postal mailing as well.

## **3 Retrieving online information within KHS**

After the "Internet went public" [Gar93] some of the major online hosts, like Datastar, Dialog or STN-International, announced the possibility of reaching them through Internet [Kea93] with line-oriented Telnet connections. Thus users have the chance to use the Internet instead of a public or scientific X.25 network. It may depend on local configuration, accessibility, or costs, which network they prefer. In the following we will show how to access external databases from within KHS and how to integrate search results into the hypertext. As examples we will discuss retrieval with STN-International and searching in our university library.

If a user browsing in KHS needs more information about a specific topic, he can access external online databases to satisfy his information need. During online retrieval KHS will give support for all steps handling the access to the external information sources. The processes of an online search includes the preparation of the search in KHS, the search in external databases, integration of the retrieved information into KHS, and access to full text.

### **3.1 Preparing the search**

Preparing an online search requires the formulation of a query with Boolean operators. First the user must collect terms to represent the information need. He can do this in several ways:

- select the terms while browsing through the KHS;
- use the index terms of one or several known units;
- select a passage of text. KHS will eliminate stop-words and reduce the terms of this passage to standard form;
- use the KHS thesaurus ( in the future );

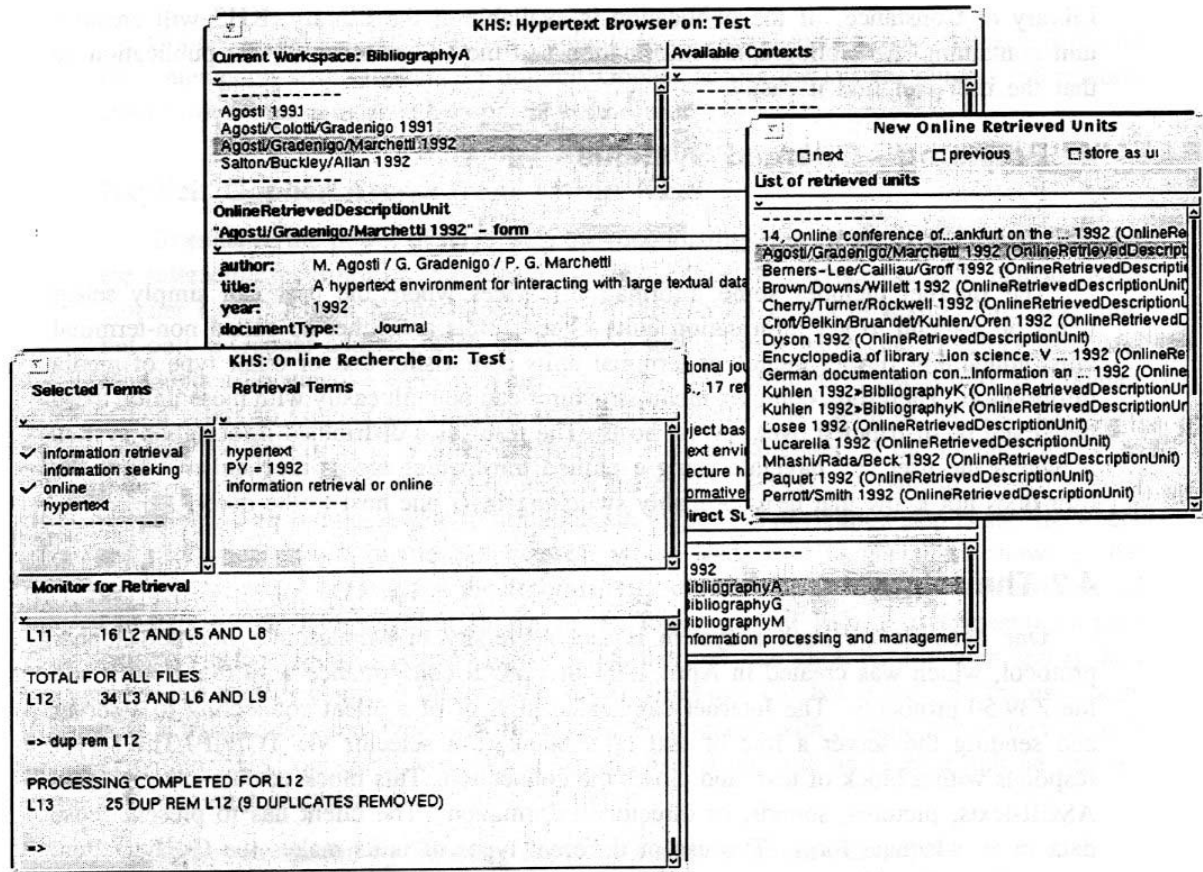
One can also search for author or get an update with an older query formulation.

In a special Online-Browser (left part of Figure 1) the user must arrange the terms he has selected into a Boolean query formulation. He combines the terms which are synonymous or similar and puts them into the right upper part of the window shown in Figure 1. In addition to the subject terms selected the user can add supplementary terms and choose several formal parameters such as a publication year, language, or document type. The user gets several lines of terms or formal parameters for the online retrieval presented at the right side of the Online-

Browser, where each line represents a block of terms combined by 'or'. The resulting lines are combined with 'and'.

### 3.2 The online search

The online search runs automatically and KHS displays the protocol of the retrieval dialog in the lower part of the Online-Browser in Figure 1. First KHS establishes the connection to the host, in our case to STN, selects the databases to search [HR93] and enters the query formulation composed by the user. After elimination of duplicates the answers will be printed and sent to KHS. KHS will then terminate the connection to the host.



**Figure 1** The Online Browser, from which the online search is handled, can be seen on the left part of the figure. On the right side a small browser shows the retrieved units with one unit selected. The selected unit within it's background is visible in the background of the figure.

### 3.3 Integration of results

KHS parses the document records retrieved, and each document record will be assigned to a new hypertext unit in KHS unless a record for the document already exists. The new units will be grouped under different aggregated nodes and contexts: First, each online search will be documented in a special unit which contains the query formulation. All the retrieved units will be linked to this unit. Second, all new units are added to existing or newly formed groups according to author, publication year, and publication type (e.g. a group for all dissertations). The user can also define groups, each group containing documents from a conference or a Journal (e.g. International Online Meeting, Journal of Documentation, etc.). New units retrieved in a search will be to these groups. KHS presents all retrieved units to the user as a list. From this list all units can be deleted which shall not be kept within the hypertext (Figure 1).

### **3.4 Access to full text**

After the online search in bibliographic databases the need for the full text documents arises. Though the availability of full text documents increases, nevertheless most of today's scientific publications are not (yet) available in an electronic form. So the user must still go to the library and borrow a book he is interested in. To support the user in finding the book in his local library, KHS can search for a publication in the University Library of Constance. If the publication is available at the Library, KHS will create a unit containing the bibliographic information and the call number of this publication so that the user can find it easily.

## **4 Browsing - oriented systems**

### **4.1 Introduction**

The Internet includes some information-services where the user can simply select items on a menu to get information units. These units are either so called non-terminal units containing further menus or terminal units comprising text or other type of media information. Therefore, complex menu-structures can be built easily with these units; these may even be distributed over several hosts. The result is a distributed information system in which the user can navigate using a unified transparent access mechanism. Thus the user does not know that he is probably switching from one host to the other.

### **4.2 The Internet Gopher**

One example for such a system is the University of Minnesota's Internet Gopher protocol, which was created in April 1991 in general conformance with the TCP/IP and the Z39.50 protocols. The Internet Gopher consists of a client connecting to a server and sending the server a line of text or a blank as a selector via TCP/IP. The server responds with a block of text, and closes the connection. This block of data can represent ASCII-texts, pictures, sounds, or directory information. The client has to present those data in an adequate form. The use of different types of units makes the Gopher client easy and comfortable to use, because the user needs not care about the specific operations necessary to display different types of information. Gopher directory information contains also the Gopher access information of each directory entry. This access information is not displayed but it is used to initiate a further Gopher access when the user clicks on an item in the Gopher interface. The access information can point e.g. to another host elsewhere in the network. Thus the connections between Gopher servers are virtual. Once the Gopher client has connected to a Gopher server, the user can browse through information on any additional Gopher servers that are connected to the first server. [Kro92][Tho92]

If the user selects a unit which stands for an index-search, the user is asked for a search term. Thus, the browsing paradigm is not exclusive within Gopher, and the user is able to search on a number of different index-based Gopher servers such as a Veronica server (Very Easy Rodent-Oriented Net-wide index to Computerized Archives) or phonebook Servers (CSO/"qi"). Veronica servers have an index containing the titles of all selectable items of all Gopher servers.

The Internet Gopher protocol was created in order to implement an easy-to-handle and distributed campus-information system based on a client-server architecture. Since 1991, the number of servers and the number of users operating clients has grown steadily and at most universities in the industrial nations Gopher servers have been established.

### 4.3 Access to Gopher within KHS

Gopher access can be explicit or implicit. In explicit Gopher access, the user is placed into interaction with Gopher. In implicit Gopher access the system obtains information from Gopher in a manner transparent to the user.

#### Explicit Gopher access from within KHS

To enhance the power of KHS with the functionality of a Gopher client and to enlarge the interactive possibilities, we created a special type of a hypertext unit, named 'KHS Gopher Unit'. A unit of that type does not represent a media or textual hypertext unit, but only a Gopher address, which contains the specific host name, the directory-path, and the Gopher-Document's type-id. Thus, a Gopher unit within KHS exists only as a pointer which refers to Gopher information across the Internet. As soon as the user navigates through the hypertext to a KHS Gopher unit, the KHS connects the Gopher server in order to get data. The user deals with a transparent access mechanism, which always guarantees the most up-to-date information. In Figure 2 a navigation step is shown, where the user selects a unit named: "Lectures" which leads to a Gopher unit showing the current university catalogue for the Information Science Department at the University of Constance. The Gopher unit shown in the lower part of Figure 2 represents Gopher directory-information. In these Gopher units the user can click on any item and is then guided to another chunk of information anywhere in the 'Gopher space'. So the user deals with a transparent access mechanism because he interacts with a Gopher directory as well as with a KHS composite unit. Whenever the user deals with a Gopher unit he can use the following features:

- Gopher-navigation by selecting items by a simple mouse-click;
- a step-by-step backtracking function for the Gopher navigation;
- a function to ascend within the Gopher server's directory-path;

A hypertext user creating a new KHS Gopher unit must define which Gopher unit from which Gopher server should be loaded when the unit is selected during the navigation. Therefore an input mask is needed, where some settings must be specified:

- The hosts Internet name-address;
- the hosts TCP/IP port number; . r
- the hosts directory-path;
- the document type expected;

To enhance this cumbersome procedure we propose a solution where the author can browse through the Gopher-space from within KHS in order to integrate any suitable chunk of information into his current hypertext-context. The user can navigate by clicking on a KHS Gopher unit's text surface through the 'Gopher-space' in the same way as when using a conventional Gopher client program. In practice, the user defining a new Gopher unit is connected to a default Gopher server and gets its root-menu. After this, he navigates through the 'Gopher space' in order to reach the Gopher selection wanted. The last action the user must perform is to fix the current Gopher selection into the hypertext unit he has just created. At that moment KHS finally memorizes the Gopher unit's address-information. In this way, documents obtained through Gopher interaction may be embedded into the poly-hierarchical structure of KHS. Therefore the user has access to those Gopher units embedded in the same way as access to all other KHS hypertext units left. This allows the user to organize his Gopher data-resources in his own and individual way for his convenient use.



Whenever the user selects a KHS Gopher unit a Gopher access is performed. If the user trusts in the maintenance of the Gopher information available, it would be useful for him to integrate such Gopher information to his KHS-hypertext. So the user has the guarantee to get the most up-to-date information.

Some principles which are typical for hypertext systems, such as hot-word links or 'guided tours', are also useful when exploring Gopher within hypertext. Browsing KHS, the user is able to define links starting from or pointing to Gopher units as it is usual for all the other unit-types within KHS. So Gopher information can be part of the user's private, group, or overall hypertext environment. For example, it is possible to generate annotations to Gopher units or to define links from self generated units to Gopher units (e.g. for serving the purpose to illustrate something).

### **Implicit Gopher access from within KHS**

The features described above are concerning the possibilities of directly imbedded Gopher information units to a KHS hypertext. An example of another application for which the Gopher protocol also is useful, is the integration of special Gopher searching capabilities such as an access to a CSO ("qi") phonebook server activated by a simple mouse-click.

On those phonebook servers information about persons such as their phone number, their room number and their e-mail address, are available. To search for any person, e.g. at an university in the US, a Gopher user must navigate to a Gopher directory containing all the phonebooks available in the world, must manually search for the right phonebook and click on it. A dialog window will be opened, where the user can retrieve data about the person he is searching for. In the KHS, the procedure which has been described is very simple as the user gets the information performing a simple mouse-click.

Assume, the user wants to fill out an address unit (the type "address unit" is a subtype of "form unit"), but he only knows a name and an organisation, then, after he typed in name and organisation and started the search, KHS performs a search for an adequate phonebook server and after being connected it retrieves the person's data and integrates it into the form. The information about an adequate phonebook server is taken from that Gopher directory containing all the phonebooks available. If more than one organisation's name matches the query, the user is prompted to select a single one. KHS stores the information obtained to that address unit.

## **4.4 The future**

### **KHS as a Gopher server**

One aspect of 'open systems' is to have connections from a local system to other information resources or applications. KHS realizes these connections using a computer network, but they are unidirectional. KHS can look to the outside, but other systems cannot look into KHS. It would be an improvement to do that. To deal with that question we are planning to implement a special Gopher server, which is attached to KHS. This special Gopher server would allow any Gopher users elsewhere in the world to operate on special ranges on any KHS hypertext assigned to public use. In the long run, the WWW (World Wide Web) protocol, which has been derived from the Internet Gopher protocol, could also represent those links, which originally had been established under KHS.

### **KHS as an authoring-tool for Gopher servers**

Furthermore, KHS will be enhanced by a feature to use it as an authoring tool for Gopher server administrators. The administrators will be able to convert KHS menu structures

consisting of terminal nodes and non-terminal nodes into the simple file/directory structure the Gopher server requires.

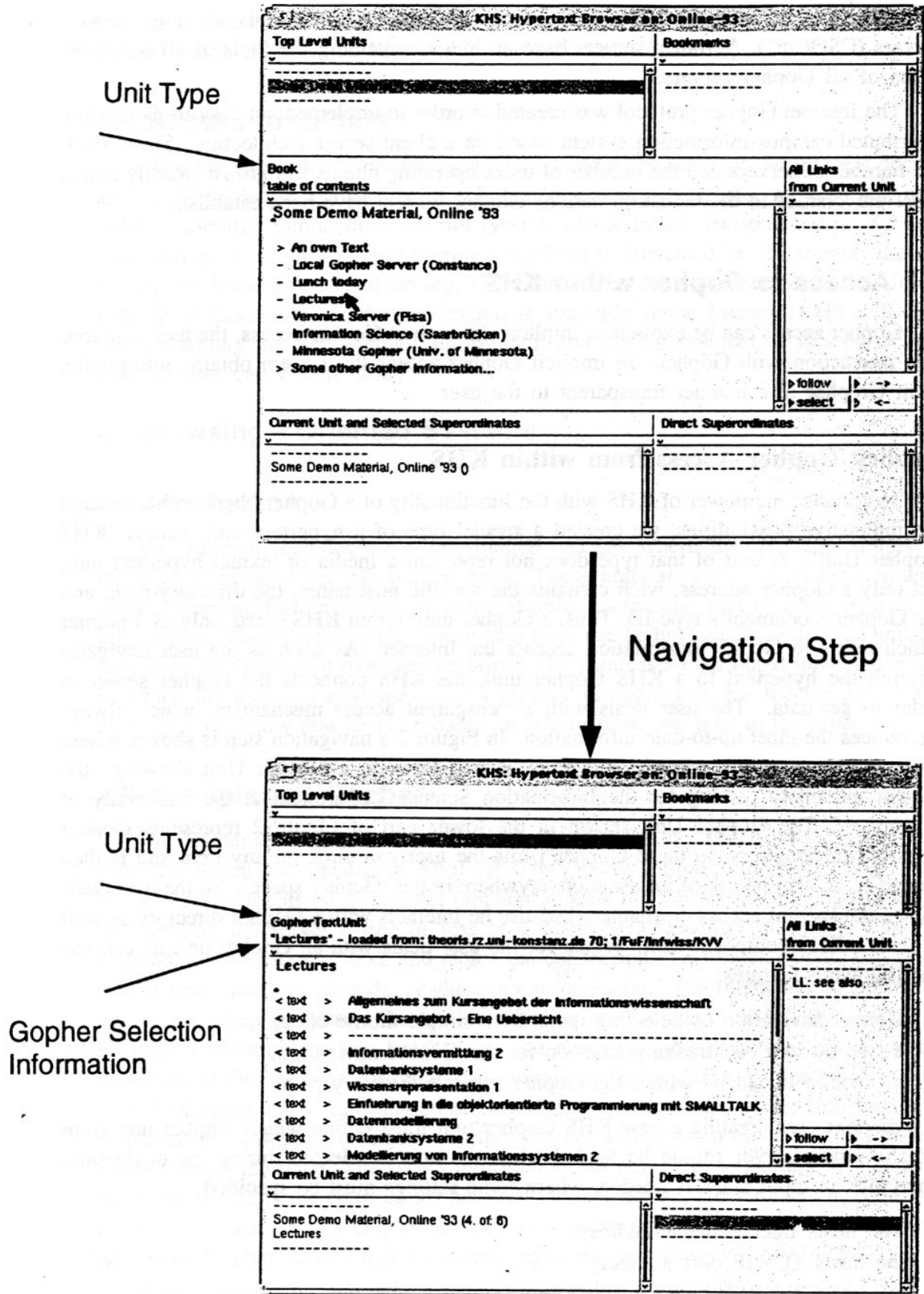


Figure 2 A navigation step towards a KHS Gopher unit. The user deals with the KHS user interface and gets transparent access to the current Gopher university catalogue.

## 5 Conclusions and outlook

Communication possibilities through the Internet have been introduced, based on the open hypertext-system KHS. The differing interaction styles of e-mail, online databases and Gopher have been integrated in KHS, so that one can communicate with the mentioned information sources using an uniform interface. This simplifies and improves the user's interaction and gives access to huge volume of external information. Through KHS information from the different information sources is accessible, so that the user can navigate through e-mail-, bibliographic or Gopher units, and link the units according to his personal requirements. Though automatic linking is supported in KHS, we will insert more sophisticated techniques, like knowledge-based or statistical procedures, to assist the user more reasonably during navigation and browsing. Furthermore we plan to combine KHS with an object-oriented database system to support multi-user access and improve the integration of the external information sources. An implementation of a Gopher server is planned, which will allow any Gopher user elsewhere to operate on special ranges assigned to public use, which originally had been established under KHS.

## References

- [Gar93] N. Garman: The time is now: the internet goes public! Online, 1, 8-9, 1993.
- [HR93] R. Hammwöhner, M. Rittberger: KHS - Ein offenes Hypertext-System Bericht 28-93 (WTTH-3/93). Technical report, University of Constance, Information Science, 5/1993.
- [Kea93] T. Keays: Searching online database services over the Internet. Online, 1, 29-33, 1993.
- [Kro92] E. Krol: The whole Internet: Catalog & User's Guide. O'Reilly Associates, Inc., 1992.
- [Tho92] C. Thomer: Information technology standards for libraries. Journal of the American Society of Information Science (JASIS), 9/92, 556-570, 1992.