A HYPERTEXT KNOWLEDGE BASE FOR PRIMARY CARE - LIMEDS in LINCKS.

Toomas Timpka[1,2] Lin Padgham[1] Per Hedblom[1] Stefan Wallin[1] Gösta Tibblin[3]

[1] **Department of Computer Science** [2] **Department of Community Medicine**
Linköping University, 581 83 Linköping, Sweden
[3] **Department of Family Medicine, Uppsala University**

Abstract

In organized health care, primary care is the first level. It is characterized by the wide span of health problems managed as well as remote location from traditional medical information and knowledge sources. The LIMEDS project has formulated the special requirements for integrated knowledge and data base management in primary care. This paper presents *Gösta's book*, a hypertext knowledge base implemented in LINCKS, an object oriented, networked database system. Firstly, aspects which make integrated hypermedia systems particularly suitable for application in primary health care are explored. We then describe the hypertext knowledge base, consisting of 500 basic text objects and 3000 links, and current implementations using the NODE data model. NODE is implemented on a SUN III fileserver, and the user interface for the hypertext context on Apple Macintosh (TM). Combination of design methods towards a parallel means-ends strategy was found to be necessary to achieve *Gösta's book*. Design groups need to be composed of computer science, medical, psychological and organizational competences.

Introduction

In a recent survey, two of three Swedish general practitioners (GPs) perceived the overall supply of medical information as less than satisfactory or unsatisfactory [1]. Four of five GPs experienced major hindrances in information searches. However, traditional information sources, such as textbooks and consultation with colleagues, were still sufficient to completely resolve every second information dilemma, even though a straining loss of practice time was perceived. Of the remaining dilemmas that were not resolved, differential diagnosis was most frequent, a dilemma for which no traditional reference source is ready at hand today. Results like these imply a need for renewal and reorganization of information facilities in primary health care.

The hypermedia idea responds in an interesting way to this unsatisfactory situation [cf. 2]. Traditional compartmentalization of subject areas and standardization of information access sequences are severe impediments for use of information resources by GPs. In many relevant medical areas information is, moreover, mainly available in non-textual form, for instance use of images in dermatology and graphs in clinical physiology.

The investigation by the LIMEDS-LINCKS project into the possibility of using computers as mediators of daily information for GP's led to conclusions regarding the necessity of an integrated design [3]. A hypermedia system is envisaged with the following contexts (sets of one or more system tools or functions that are presented together for reasons of functionality or interdependency [4]):
- primary care hypertext, i.e., texts on medical diagnosis and therapy customized with regard to both structure and content;
- browsing and information retrieval from medical bibliographic databases;
- integrated electronic mail for communication with

colleagues;
- knowledge based support of differential diagnosis; and
- planning and critiquing support of therapy decisions.

In this paper the presentation is focused on the hypertext context. The aim is to present:
- the underlying database structure, and
- a hypertext for diagnosis support in primary care.
Major hypertext systems in medicine have earlier been reviewed [5, cf. also 6].

1. Structure and content of hypertext for primary care.

In order to be superior to traditional medical textbooks, a hypertext source for GPs should be enhanced in both structure and content. The need for an "open" structure follows from the fact that the information is to be used for purposes determined by the user, not the author. It should be possible to answer questions like *Of which diseases is the "red eye" a symptom of?* as well as *What symptoms has Addison's disease?*. Moreover, a GP often needs to retrieve complex, compound information which is facilitated by a non-sequential structure not found in traditional texts [7,8]. A third motivation for non-linear structure is support for browsing. Practitioners often find it difficult to formulate their information dilemmas explicitly, for example using Boolean expressions in data base queries [9]. To find what is needed through associative navigation forms an attractive alternative.

The content should be customized since the GP's work differs from that of specialist physicians in at least four ways [cf. 10]:

1. Practice is usually remote from traditional information resources such as medical libraries.

2. The GP has to be available for all types of medical problems. Consequently there is a need for a wide span of cross-referenced medical knowledge.

3. Illness is often seen early before the classical 'textbook' picture has developed.

4. The illness is unstructured, as it has not been previously assessed by other physicians. Social and psychological matters are usually important.

Thus, information resources directed towards specialist physicians are of lesser interest for general practitioners.

2. Requirements on the database system.

During development of the functional design of the LIMEDS hypermedia system for practitioners, we have developed a set of requirements which the core database must meet. Firstly, it should be possible to support the database by 'intelligent services' as an aid for computer-naive users, on an operating system level as well as in applications. This requires the representation format in the database to be sufficiently structured for computer interpretation and concurrently allow flexible human use. Secondly, it has to be possible to represent historical information *of* the database *in* the database. On a system level simple database requests from physicians and nurses often have to do with information about past as well as present database states and transactions (e.g. *Which article was it Dr. Andersson sent me last week?*). Also in order to allow the user to make individual annotations and changes to their particular version, historical information is necessary. On a higher application level historical information can even be essential, for example in planning of diabetes mellitus management.

The remaining requirements which have influenced the evolution of the low level database structure, but that are only described to a lesser extent in this paper are:
- integration at the representation level, of both knowledge and data to be used for different purposes;
- high level of distribution, up to portability of parts of the knowledge base;
- support of co-ordinated multi-user work;
- communication interfaces to different patient database systems;
- readiness for transfer of knowledge to and from other medical decision support systems.

3. LINCKS

LINCKS can be described as an information management system built with the intention of integrating intelligent services and connected to an object oriented, networked database with history facilities [11].

3.1 Database structure.

The database in LINCKS, NODE, is object oriented, in that it supports the representation of real world objects, directly as database objects, possibly structured objects where each part is in itself an object. For instance, a document can be represented as a structure of sections and paragraphs, a filing cabinet as a structure of files, which are themselves structures of documents, etc.

The basic data structure in NODE (called a node) from which all objects are built up, has three distinct parts [12]:

* An *image*.
* An *attributes* section
* A *links* section

The image section is intended to contain information which is meaningful for the user but is not intended to be interpreted by the system. The image can be free text, graphics, sound, animation, etc. In the primary care hypertext context described here the image contains the text with the *source points* for *reference links.*

The attributes of a node contain structured information about the node which may or may not be interesting for the user, but most importantly is sufficiently structured that it can be interpreted by application or LINCKS system software. Simple attributes which are useful in hypertext applications are keywords, author name, dates created and updated etc. The attributes can, however, also be used for describing objects in expert-system-type applications. The underlying database structure places no limitations on number of attributes or type of attribute values (other than that attribute values cannot be a direct link).

The links section is used to store pointers from one object to others. It is these pointers which are used to implement the hypertext functionality of being able to 'jump' from some marked place in the text, to other text(s) (paragraph(s)) (Figure 1). The structure of the links section is three layered. The first layer of structure is a group name, the second layer a field name, and the third is an ordered list of the (logical) pointers to objects. As with the attributes the database places no limitations on number of groups, number of fields in a group, or number of link pointers in a field value. The only limitation is that each group name may occur only once within each object, and each field name only once within each group.

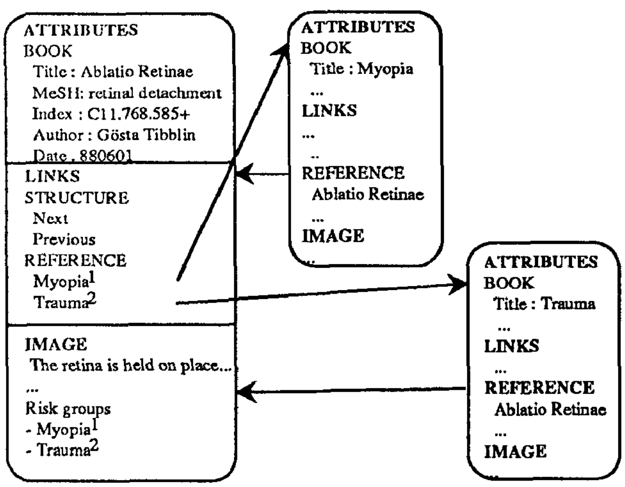

Figure 1. Representation of the primary care hypertext in NODE.

In the primary care hypertext application, the link group names have been used to separate out two different types of links: *reference links* and *structure links.* Structure links represent the linearity of the document ('next paragraph', 'next chapter', etc.) while reference links represent cross-references (definitions, 'see', 'see also', etc.). It is likely that further development will lead to identification of other link types which can then be represented as particular groups. The field name is used for the user to attach a name to the link. If no name is attached by the user, the link name is given the value of the name attribute of the destination object. The ordered list is not used more than that it allows more than one outgoing link with the same name, if desired.

3.2 Management of information history.

A link can be a pointer to an object without reference to time, in which case when the link is followed one will obtain the current version of the object, or it can be a pointer to an object as it existed at a particular time. The 'particular times' represented in the database are the end of each work session (measured by a logout from the system) plus any time points at which a user has explicitly stored to the database during a worksession.

Three types of history information have been identified, as important within the LINCKS system.

Object history is the history of an object as it has developed over time. It enables reconstruction of the object as it was at any past time it was saved in the database. The information within this type of history can be used in the answering of queries such as: *What was last year's version of the diabetes management recommendations?*

Edit history is the history of what was used as a base for particular version of an object. When development of an object is consecutive, edit and object history are identical, as the basis for a new version is then the previous version. However, it is not uncommon, particularly in document preparation, to use something other than the most recent version as a base. Text or formatting attributes for example may be taken from some other document, or from some earlier, now preferred version of the document.

Both edit and object history exist at all levels - the edit history of a particular paragraph can be retrieved as easily as the edit history of the document as a whole

Command history is the information about what actions have been done on what objects, and by whom. The command history is not yet implemented in NODE, as it is dependent on the user interface which up till now has been only temporary. It is planned to be implemented as a structure of user actions, partially ordered over time, with links to those objects affected by the actions.

3.3 Implementations

LINCKS is today implemented in C under UNIX using SUN III as a fileserver and one or more of a number of machines as workstations -- currently Sun III, Xerox 1186, Apple Macintosh, and DG1 laptop portable computers.

The implementation of the primary care hypertext context has three program layers: The outermost layer is the user interface which is so far only implemented on the Apple Macintosh with supplementary Radius 18 inch screen. It is implemented in ObjectPascal, using the MacApp object-oriented software framework [13]. It communicates with the Macintosh interface regarding such things as mouse input, window and menu management (Figure 2). The second layer manages application specific attributes such as keywords, and link types specific to the primary care hypertext (structure and reference links). The third layer manages the database functionality [12], including such things as connecting *source points* in the text to the actual outgoing links, and filtering the image text so that markers for link *source points* are not seen. The database layer also deals with such things as structure management, history, read/write, concurrency control, etc.

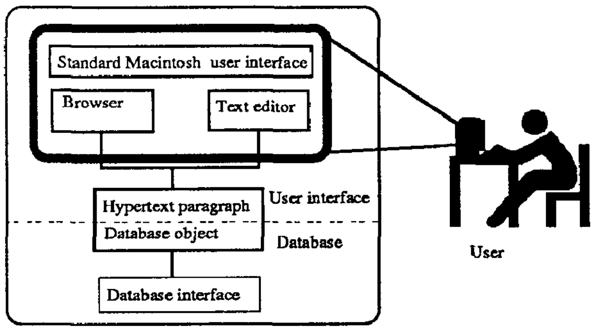

Figure 2. Relation between user interface software and the NODE database.

4. Gösta's book: A primary care hypertext.

The core for the hypertext context consists of a primary care knowledge base, *Gösta's book*, compiled by the faculty at the Department of Family Medicine at Uppsala University. (All texts are in swedish, the paragraphs used in the figures of this paper have been translated separately). *Gösta's book* can be described from three aspects:

- The knowledge base;
- Access strategies; and
- Creation of new hypertext objects.

4.1 The hypertext knowledge base.

The knowledge base is composed of 500 basic text objects describing medical symptoms, signs, diseases, laboratory tests and treatments connected by 3000 links (Figure 3). Organization is based on structuring objects, such as library, book, document, index and problem area abstract (see below). All descriptions are specially adjusted for use by GPs.

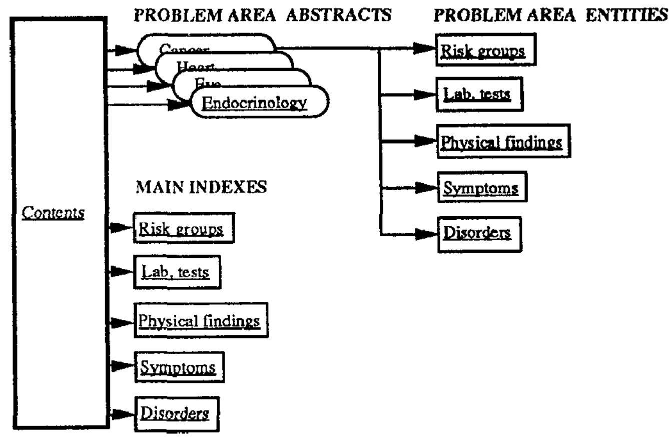

Figure 3. Logical structure of the knowledge base in *Gösta's book.*

4.2 Access strategies.

Access strategy is determined by the practitioner. When the cursor is moved over a text containing a *source point* then the cursor displays the number of outgoing links. Mousing the item displays the names of the links, and the desired destination can be selected.

4.2.1 Indexes.

Firstly there are menu-based indexes from symptoms (755 items), signs (415 items), laboratory tests (255 items) and 'risk groups' (320 items) to the 265 major disease categories. The indexes give direct access to manifestation descriptions, which first define the manifestation and then give a listing of diseases where the manifestation can occur grouped by prevalence and severity. From each listed disease a link can be followed to an overview of the corresponding disease (Figure 4).

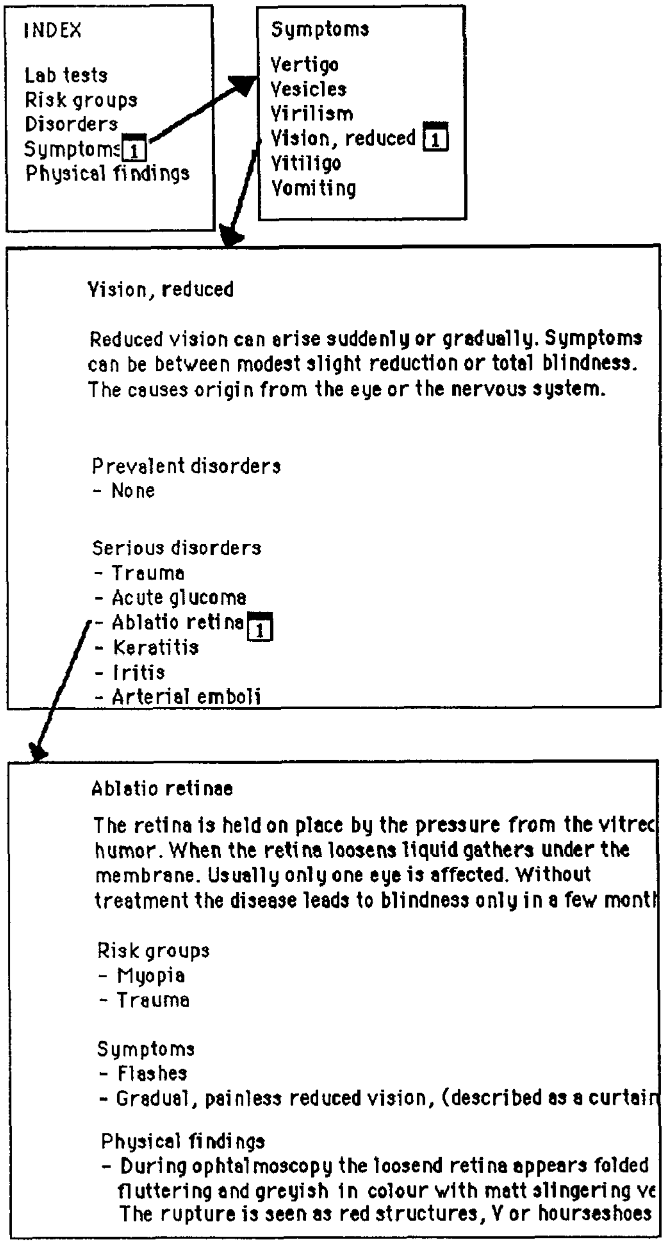

Figure 4. The navigation sequence for reaching a disease description from a manifestation index. The number of outgoing links from a *source point* is displayed in the cursor.

The disease overviews describe the disorder with a short textual overview and a listing of risk groups, symptoms, signs and laboratory tests. From short therapy descriptions, links to more detailed therapy recommendations are available. In the disease overview text, cross-reference links are also available regarding differential diagnosis, prognosis and special considerations (Figure 5).

4.2.2 Problem area abstracts.

Secondly, the information can be accessed by 22 problem area abstracts, for instance *Endocrinology* (c.f. Figure 3). These abstracts describe the interrelation between diseases and disease manifestations in a particular problem area with regard to e.g. prevalence, incidence and severity. The text in the abstracts is cross-referenced to corresponding disease overviews as well as to the specific descriptions of symptoms, laboratory tests, etc.

4.2.3 Paths.

Thirdly, a "path" can be chosen from a graphical browser available in a separate window (Figure 6). Paths supplied by the author can be traversed (for example, *tricky issues diagnosing a red eye*), but the individual practitioner can also save his own paths for later use. The browser is always available to dynamically display the path traversed during a session.

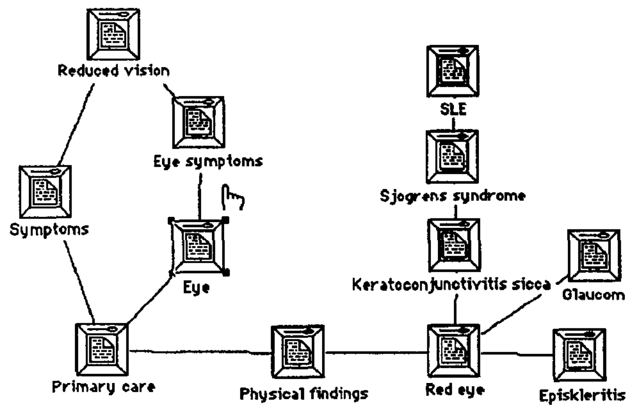

Figure 6. A dynamic graphical browser is available in *Gösta's book*. Paths can be stored, and used in a later session.

4.2.4 External database access.

A fourth way of using the system is as an interface to external databases, primarily MedLine. All disease objects have MeSH (Medical Subject Headings) terms in their attributes. Keywords of traversed objects during a session are saved and can be used as a first template in the formulation of a search query to MedLine. The search query is then formulated in a separate query formulation context.

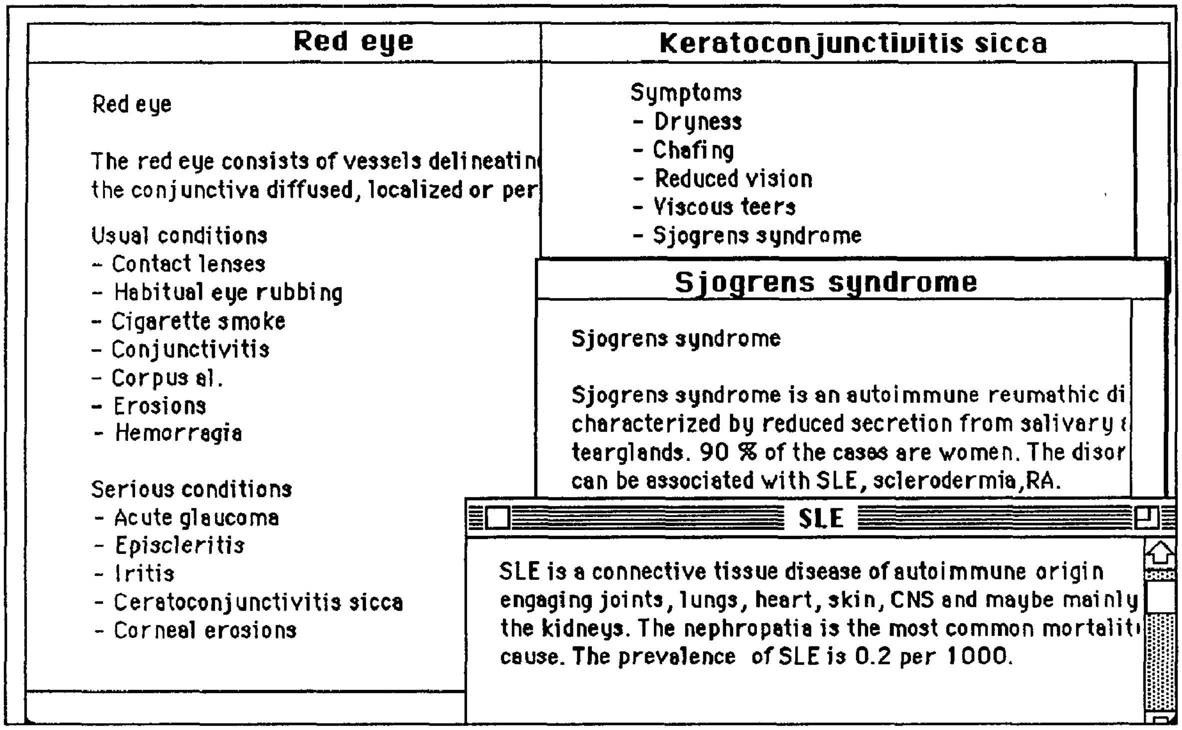

Figure 5. Problem areas are frequently cross-referenced in *Gösta's book*, here ophtalmology and rheumatology.

4.3 Creation of new objects.

Creation of new objects and links is performed interactively, by direct manipulation (Figure 7). To create a new link first a *source point* is determined. Then the link is drawn *from* the *destination object*. The user is prompted to give a name for the link. If no name is given, the link name is given the value of the *name attribute* of the object pointed to. To create a new text object, for example an annotation, firstly an object with empty image is created. Text is thereafter entered into the image, and the object is connected to the knowledge base by drawing a link to the desired *source point*.

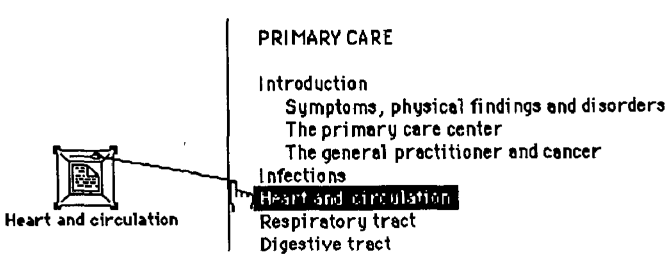

Figure 7. Creation of a link in *Gösta's book.* A *source point* is determined, thereafter the desired object is connected by, literaly, drawing the link.

Discussion

5. The LIMEDS-LINCKS perspective.

LINCKS-LIMEDS has been developed in response to the deficiencies of traditional database management systems as seen from the perspective of authentic needs in primary care. It is thus significantly different from earlier information management systems in general and hypermedia systems for application in the medical field in particular.

5.1 Representation format.

The object-oriented basic data structure in NODE with its free-format image, machine-readable attributes, and the possibility for unlimited number of types of links, is a good base for hypertext implementations. By adding relatively simple attributes and system functions one can significantly increase the functionality of the system. For instance, one could have the system automatically add attributes to mark when a user annotated an object; the system could then easily retrieve all objects annotated by a particular person. NODE also provides the basis for software which must interpret the information in the database substansially more, e.g., software for drug prescription support. The expressiveness of NODE is demonstrated by that it is straight-forward to map to NODE from HAM (The Hypertext Abstract Machine [14]) and thus also from well-known systems as *Guide, Intermedia and Notecards*.

However, the way to define the connection between *source points* in LINCKS-LIMEDS' *image* text and *links* differs from HAM in that the pointers are not stored in link attributes and pointing at bytes in the *image*, but are instead stored in the *image* pointing at links (cf. blocks in Intermedia [15]). Thus, text and *source points* can easily be removed and added to the *image* without displacements.

5.2 The relation to relational databases.

The development of NODE was an attempt to find a database structure on level of representation that was sufficiently expressive that it would be adequate to represent the wide variety of objects and relations found in the complex medical environment, but also sufficiently structured that it could be interpreted by appropriate software. Traditional relational databases are characterized by data independence, i.e., a barrier between data and the procedures working on them [16]. In LINCKS no such distinction exists, since actions (procedures) are also objects. However, connections between the object-oriented paradigm, relational databases and intelligent support has earlier been approached as an object-based interface between relational databases and expert systems. In PENGUIN [17], a structural model is used to generate an object schema (with ownership, reference and subset link types) from the traditional data definition language. This object schema can then be used for reasoning and navigation. However, since the object schema is not stored but has to be computed dynamically, the approach is less interesting for applications where instant link traversal is essential, such as hypertext.

5.3 Hypertext in a multi-user environment.

Another difference comparing LINCKS to many hypertext systems is the ability to have a history of multiple parallel versions of an object, which is a desirable, or perhaps even necessary, feature for hypertext systems in multi-user environments. This feature makes it possible for individual users to make their own annotations to generally distributed documents. If a practitioner has a contradictory opinion of a fact in the knowledge base it is often preferable to keep the contradiction, to let the user make an annotation (or modify their personal version), rather than to modify a single version. Today it is only possible to make personal annotations to *Gösta's book*, not to own and modify a personal version.

5.4 Hypertext vs. Expert systems.

LINCKS-LIMEDS can also be compared to expert system 'shells', in which the system provide structures for building up a specialized specific purpose data(knowledge)base and tools for problem-oriented information access. However, these systems often have poor communication and maintenance facilities in the database and a computer-controlled interface dialogue. Survey studies of practitioner attitudes have also shown preference for fulltext information retrieval rather than expert systems [9].

5.5 Application of hypermedia systems in medicine.

Hypermedia applications in the medical field inherit the advantages, but also the problems, in general hypermedia research. Some issues are, however, specific.

5.5.1 Searching Gösta's book for information.

Orthogonal to the suggestion that each text object in a medical *hyperlibrary* should be searchable [18], searches are in *Gösta's book* envisioned on document level. Our approach to retrieving information from *Gösta's book* is instead navigation, using the document semantics. Within books and documents, navigation is enhanced by structuring objects, such as the problem area abstracts. Navigation is also enhanced by that the link/object ratio in *Gösta's book* is high, approximately 5.5. This ratio can be compared with corresponding ratios from online manuals, for example 2.3 for the online manual available for the Symbolics Lisp-machine [19]. The keywords used to index individual text objects are used for generating information searches outside the document, e.g. collections of books and documents in LINCKS or external databases. It would of course be possible to extend *Gösta's book* to provide within-document search using these keywords. However, navigation at this level has so far proved satisfactory or even preferable to search.

5.5.2 Indexing of medical knowledge.

It is also important within a medical hypermedia system to have a common terminology for the knowledge to be used. A problem with indexing *Gösta's book* was that the present knowledge organization systems are only useful for one specific function (cf. MeSH for bibliographic use, as mapping between disorders and their manifestations are missing). However, combination of MeSH with clinically oriented organization systems (CMIT, SNOMED) has proven promising [20].

5.5.3 Knowledge support for primary health care.

Gösta's book and LIMEDS-LINCKS can be compared with The Oxford System of Medicine (OSM), which is an extensive knowledge based system under development aimed for primary care [21]. In OSM much attention has been paid to the user interface, by which access is supplied to an integrated patient record, support for diagnostic analysis and choice of therapy. The core representation format is a PROLOG-like language, PROPS-2, since the focus is on the expert system component of the multi-function system. Expert system reasoning is, however, difficult to apply for reasoning in primary care since many problems have psycho-social aspects which are hard to represent in logic [9]. In LIMEDS the focus is therefore on full-text information retrieval, mainly by hypertext methods, although the NODE data structure also allows integrated machine-readable representation of knowledge (see also 6.1.). The advantage with *free-format* representation is that knowledge can directly be entered accessed and cross-referenced, without translation into formal language or paying attention to the *closed world assumption.* In the therapy planning and drug prescription context, however, the knowledge of counterindications and side effects is well enough defined that LIMEDS mainly makes use of it in *machine accessible* form.

5.6 The key to the future: multidisciplinarity.

In the LIMEDS-LINCKS project we have concluded that to develop the needed new information facilities for primary health care, new methods have to be explored. Within earlier development of decision support systems in medicine two general methods can be recognized: the database centered [cf. 22], and the user-situation centered [cf. 23]. We have instead found it necessary to combine these methods towards a parallel means-ends strategy. Required organizational changes and, for instance, design of user interface cannot be seen isolated from the database design and vice versa. This requires composition of design groups with computer science, medical, psychological and organizational competence. It is in such a setting the LIMEDS-LINCKS system is emerging.

ACKNOWLEDGMENTS:
This work was supported by grants from the Swedish National Board for Technical Development, the Swedish MDA-program and the Swedish Medical Research Council.

REFERENCES:

1. Timpka T, Ekström M, Bjurulf P. Information needs and information seeking behavior in Primary Health Care. Scandinavian Journal of Primary Health Care, 1989, in press.
2. Nelson TH. No more teachers' dirty looks. Computer decisions, 1970;9: 16-23.
3. Timpka T. Knowledge-based decision support for general practitioners: an integrated design. Comp Prog Biomed 1987;25:49-60.
4. Åberg P. LOOCS. Design of a User Interface Support Manager. Licenciate Thesis. Department of Computer Science. Linköping University, 1989.
5. Timpka T, Hjerppe R, Zimmer J, Ekström M. Introducing hypertext in primary health care: Supporting the doctor-patient relationship. Proc. of User-oriented Content-based Text and Image Handling '88, Cambridge MA, 1988.
6. Conklin J. Hypertext: An introduction and survey. Computer 1987;Sept: 17-41.
7. Weyer SA. Searching for information in a dynamic book. XEROX PARC Report SCG-82-1, 1982.
8. Schneiderman B. User interface design for the hyperties electronic encyclopedia. Proc. of Hypertext '87, Chapel Hill, 1987.
9. Timpka T. Introducing hypertext in primary health care: A study on the feasibility of decision support for practitioners. Computer Programs and Methods in Biomedicine, 1989, in press.
10. McWhinney IR. Problem solving and decision making in family practice. Can Fam Physician 25:1473-1477, 1979.
11. Padgham L. LINCKS. Proc. of IFIP Conference on Office Automation Systems, Elsevier, Amsterdam, 1987.
12. Padgham L. NODE. Proc. of ISMIS '88, Turino, 1988.
13. Schmucker K. MacApp: An application framework. Byte 1986;August:195-196.
14. Campell B, Goodman JM. HAM: A general-purpose hypertext abstract machine. Proc. of Hypertext '87, Chapel Hill, 1987.
15. Meyrowitz N. Intermedia: The architecture and construction of an object-oriented hypermedia system and applications framework. Proc. of OOPSLA '86, Portland, 1986.
16. Tsichiritzis DC, Neirstrasz OM. Fitting round objects into square databases. Proc. of ECOOP '88, Springer Verlag, Heidelberg, 1988.
17. Barsalou T. An object-based architecture for biomedical expert database systems. Proc. of SCAMC '88, Washington DC, 1988.
18. Frisse ME. Retrieving information from medical hypertext systems. Proc. of SCAMC '88, Washington DC, 1988.
19. Walker JM, Moon DA, Weinreb DL, McMahon M. Symbolics Genera Programming Environment. IEEE Computer 1987;20
20. Martin BK, Rada R. Building a relational database for a physician document index. Med Inform 1987;12:187-201.
21. Fox J, Glowinski A, O'Neil M. The Oxford system of medicine: A prototype information system for medicine. Proc. of First European Conference on Artificial Intelligence in Medicine, Springer Verlag, 1987.
22. Pryor TA, Gardner RM, Clayton PD, Warner HR. The HELP System. J Med Sys 1983;7:87-102.
23. Lind M, Pettersson E, Sandblad B, Schneider W. Computer based workstations in health care. In: Bakker AR, Ball MJ, Scherrer JR, Willems JL (eds.). Towards new hospital information systems. North Holland, Amsterdam, 1988.