

Information Retrieval Systems for Professionals: A Case Study of Computer Supported Legal Research

Morten Hertzum, Henrik Søes, and Erik Frøkjær
Computing Science, University of Copenhagen, Denmark

Abstract. Two design ideas concerning information retrieval systems (IRS) for professionals are evaluated. First, we aim to support a professional's document handling. Second, we aim to ease the evolution of the IRS, i.e. allowing it to be modified in a straightforward and flexible way as new requirements arise. The viability of these design ideas are illustrated through a case study concerning the development of a prototype legal IRS based on a leading text of Danish laws. Important facilities in the prototype include a dynamic thesaurus, a dynamic classification structure, and personal notes. The prototype is built using a relational database, not inverted files as in the majority of IRS.

Introduction

Information retrieval is the activity of retrieving stored data that provides the information to fill a gap in the user's "anomalous state of knowledge" (Belkin et al., 1982). We are interested in information retrieval systems (IRS) for professional use. Professionals are subject specialists in their task domain, as opposed to specialists in documentation, such as online search experts. Further, professionals are distinguished from clerks, i.e. persons employed to keep records, accounts, files, handle correspondence, or the like.

Our work has grown from two starting points: (1) We assume that information retrieval should not be treated as an isolated activity, as it is intertwined in the users' entire work situation. This calls for a high level of openness and integration among the computer systems involved. To support the user in his document handling, we see a special need for combining facilities of filing and retrieval. (2) Complex computer systems such as IRS for professionals cannot be developed once and for all. Therefore, it should be a key consideration in the design of such systems to include facilities supporting their evolution. The desired level of changeability must be obtained together with acceptable efficiency in terms of response time and space requirements; but typically the efficiency requirements are favoured at the expense of changeability (Nishimoto & Ura 1989).

Most existing IRS are based on inverted files. Inverted files offer fast retrieval, but the cost of updates is high (Faloutsos, 1985). The major reason for the popularity of inverted files seems to be that, in most IRS, updates are centralised and rare compared to retrieval. Thus, the capability to handle changes is less important than efficient retrieval. From our perspective, IRS for professionals require much changeability. We will illustrate this by discussing the design and implementation of an IRS which can evolve in a flexible way. To meet the demand for changeability we propose to base the IRS on a relational database—a database model developed with the aim of handling changes effectively (Codd, 1970).

To test our starting points we have carried out a case study in the legal area. Lawyers are professionals who have developed and refined their tools and work routines for many years. Furthermore, legal text retrieval has held a central position in information retrieval research for many years (Tenopir, 1984); and Karnovs Forlag, the publisher of a leading body of laws in Denmark, gave us access to a machine-readable version of part of one of their publications.

We developed a prototype IRS with the aim of investigating how to provide lawyers and others working with legal issues with relevant recorded knowledge. Legal work is concerned with legal norms; these norms are subject to continuous interpretation and argumentation by lawyers drawing in aspects of the social, cultural, and historical context. From this perspective we do not

find it realistic to automate this interpretation process (see Leith, 1986; Frøkjær, 1989).

First we describe Karnov's Lawbook and the process of keeping it up to date. Then we turn to the design and implementation of a prototype of Karnov's Lawbook as a computer-based legal IRS. Finally, we discuss the two major design ideas—support for document handling and evolution. We consider these ideas to be of general relevance to IRS for professionals.

Karnov's Lawbook

Karnov's Lawbook (Danish: Karnovs Lovsamling) is a printed collection of legal texts. It is the only annotated body of laws in Denmark covering almost every law currently in force (Blume, 1989; von Eyben, 1989). It consists of three parts: the legal texts, the notes, and the indexes. The legal texts are included in full with unchanged wording; interpretations and comments are given in the notes. The notes, which are made by 180 legal experts, are primarily summaries of the preparatory work and of principal cases. Even though the notes are concise, they make up half of the text in Karnov's Lawbook. Access is provided by four indexes: (1) The chronological index allows the legal texts to be reached through their date and number. (2) The alphabetical index gives access through the most important words in the titles of the texts. (3) The table of contents gives a systematic classification of current Danish laws, presented in a hierarchy. (4) The subject index comprises almost 15,000 words and is arranged as a list of main entries, each with the facility for a subordinate list of more specific entries.

The first edition of Karnov's Lawbook was published in 1924. Karnov's Lawbook, current law, and legal practice have changed considerably since then. Keeping Karnov's Lawbook up to date is time-consuming and demanding in expertise, but decisive for its success. At present this is done through a two stage process: Every third year a new and reorganized edition of the entire body of laws is published and in each of the two intervening years, a supplementary volume is published containing all new and modified legal texts and their notes. To make it possible to find the current law with respect to some legal area, a separate index volume is published annually. The index volume also includes new or altered notes to unchanged legal texts. Thus, accessing Karnov's Lawbook should be done by accessing the latest version of one of the indexes and only then turning to the actual legal text. Finally the index volume should be consulted to see whether new or altered notes exist. This access process is cumbersome and, according to the editor von Eyben (1989), it is evident that even long-standing subscribers sometimes fail to do it thoroughly.

Karnov's Law Database

In this section we discuss the design and implementation of a prototype, called Karnov's Law Database (KLD), a limited legal IRS based on Karnov's Lawbook. The purpose of a complete version of this IRS would be to provide an integrated set of tools supporting lawyers in their search for information. KLD is a full-text IRS giving access to about 4 Mb of text. This is not a huge prototype, but it does amount to 10% of the entire body of laws. Our prototype is implemented in Oracle[®], Hypercard[®], and C on a Macintosh[®] IIcx.

The process of keeping Karnov's Lawbook up to date currently interferes with the aim of making it easy and efficient to use. Part of the motivation for turning Karnov's Lawbook into a computer-based IRS is to try to avoid this problem. Furthermore, we expect that the ways of searching the body of laws can be significantly improved. To make the effort of turning Karnov's Lawbook into an IRS worthwhile, it is crucial that the resulting IRS provide a platform which is sufficiently stable and flexible to form the basis for the publication of Karnov's Lawbook for a number of years. Thus, changeability must be considered a key factor during design. Furthermore, the IRS must be relatively hardware independent and programmed using techniques likely to be supported by future software tools.

We have developed KLD in a step by step manner. First we made the nucleus of the system. Then more advanced facilities were added, one by one. We did this in an attempt to test whether a relational database has the functionality and flexibility necessary to allow extensive changes and expansions. The facilities added include a thesaurus, a classification structure, and personal notes.

The nucleus of KLD

The nucleus of KLD contains the basic retrieval technique and those parts of the system that handle storage and presentation of the documents. The retrieval technique is boolean retrieval with proximity operators and wildcards. Furthermore, the legal texts can be retrieved through two facilities analogous to the alphabetical and chronological indexes of Karnov's Lawbook; i.e. documents can be retrieved by name or by date and number. We have also included limited browsing facilities. The most important one is the ability to jump back and forth between sections of the legal texts and notes referenced. Furthermore, documents can be browsed by jumping to a specified section or to the next or previous section.

In the data model of the nucleus we decided to split the legal texts into sections and store these as undivided entities; also each note was stored as one piece of text. To guide the design of relational databases, five normal forms have been formulated (see e.g. Date, 1990; or Kent, 1983). Date (1986) admits that there is some truth in regarding normalization as optimizing for update at the expense of retrieval. As we emphasize the need for changeability, we have a strong bias in favour of satisfying the normal forms. Starting off with the intention of satisfying the normal forms would at least give us experimental evidence of the magnitude of the problem before starting to optimize. We decided that the data model should at least satisfy third normal form. The data model of the entire prototype is shown in Figure 2.

The thesaurus

In a full-text IRS, such as KLD, a facility to support the selection of query terms is of great value. Therefore the first expansion is a thesaurus covering the legal area. The thesaurus should serve three purposes: (1) Offer support during the selection of terms for full-text queries. (2) Make it possible to expand queries by adding, say, related terms. (3) Enable keyword-based retrieval.

To achieve this, the thesaurus is made up of five relations. Three relations relate terms to one another: broader terms, narrower terms, and related terms. Further, the thesaurus includes notes giving definitions of terms or remarks about their usage. These four relations are standard; the fifth is, to our knowledge, something new: This relation establishes links between documents and thesaurus terms. Through this relation the thesaurus supports keyword-based retrieval since the thesaurus terms are used as keywords pinpointing documents of special interest. References can also be made to individual sections of the legal texts, allowing more specific references. If the user wants to use the keyword facility this is done by highlighting the reference to the pinpointed document and selecting the function *See Ref*, see Figure 1.

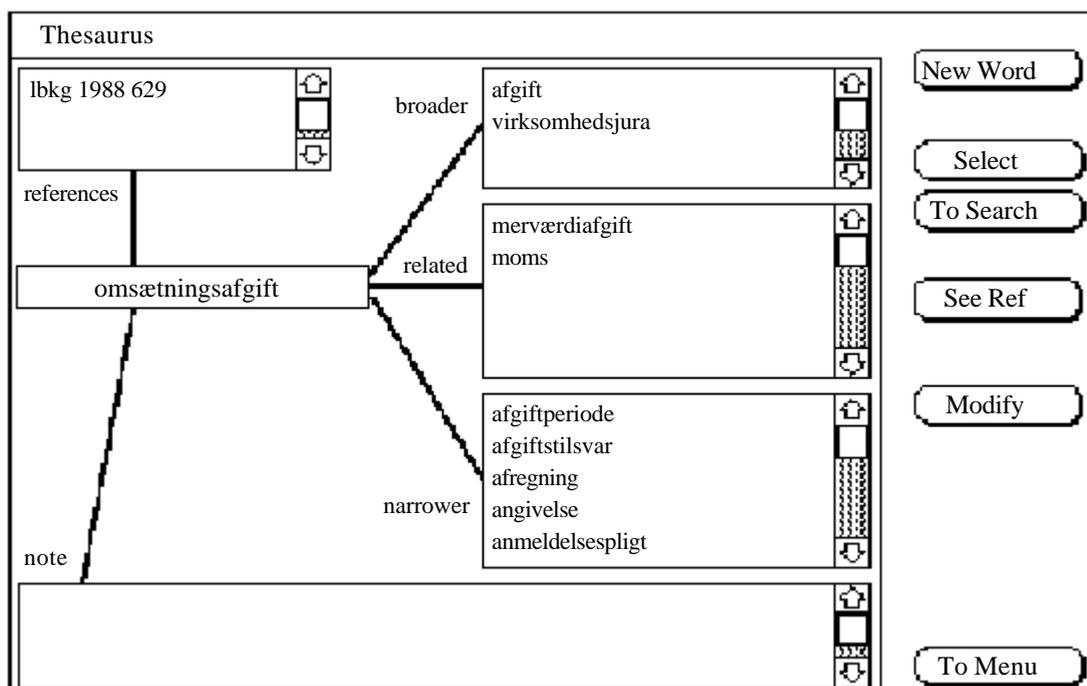


Figure 1 The thesaurus screen. The thesaurus screen focuses on one term and shows its relations to the other terms. The relations are: references making links to particularly interesting documents, broader terms, related terms, narrower terms, and notes. The available functions allow the user to move to new words in the thesaurus, to select a term for inclusion in a query, to move to the search screen, to see a referenced document, to modify the contents of the thesaurus, and to return to the main menu screen.

The value of having both full-text facilities and keywords was early emphasized by Sprowl (1981). The normal approach to indexing is to assign keywords to documents. Each document is assigned a number of keywords which together form a description of the contents of the document. We have approached indexing in another way: as documents assigned to keywords. In this sense our approach could be termed reversed indexing, i.e. documents of specific relevance to a certain keyword are pointed out, and thus directly accessible to the user. We believe placing this keyword facility in the thesaurus is well-chosen. During their search for query terms the users navigate through the thesaurus using it as a map of the terminology (Foskett, 1985); here they may profit from being presented with identifications of the documents indexed by the inspected thesaurus term.

Language usage changes all the time causing the meaning of existing terms to evolve, introducing new broader terms, replacing old terms by new ones etc. Example: The Danish word 'moms' (which translates to VAT, the abbreviation for Value Added Tax) does not occur in the legal texts defining the concept and its legal implications. It was colloquial at the time these legal texts were written. Since then the word has become established and now it appears in legal texts in place of the originally correct term. If it is not possible to incorporate alterations like this in the thesaurus as soon as the user becomes aware of them, the user will repeatedly experience the IRS as being out of date and complicating the work it was meant to support. Therefore it is of great importance that the user can make alterations to the thesaurus. From this point of view a thesaurus becomes a personal tool, not something all users must have identical copies of. In KLD the user is allowed to insert, delete, and update terms, references and notes. This means for instance that new keywords and new references between keywords and documents can be made whenever they prove to be useful.

Keeping a thesaurus up to date is typically described as a process requiring centralised control (see e.g. Soergel, 1974; Strong & Drott, 1986; Batty, 1989). Over a period of time suggestions for changes are collected; then the responsibility of bringing the thesaurus up to date is given to a central group of experts. This process is considered necessary to keep the thesaurus consistent. From our point of view such a process will often be too slow and impose unnecessary limitations on the users. For instance, many terms are used by some professionals long before the experts consider them for inclusion in the thesaurus. However, the expert group is more likely to keep the overall structure of the thesaurus in mind and remember that removal of outdated material is just as important as adding new. Thus, keeping a thesaurus up to date requires modifications made by a central group of experts as well as by individual users—or groups of users.

In KLD the individual user is allowed to modify the thesaurus and groups of users can collectively develop a thesaurus matching their needs. Modifications will only be visible to the user or group who made them. It is only the central group of experts who is authorized to modify other users' thesauri. Having two sources of modifications creates a need for a mechanism to integrate the modifications made by the user and those made by the experts. This need arises every time the experts release a new version of the thesaurus. Hertzum & Sjøes (1992) have done some preliminary work on this problem, but further research is required.

One application of modifying the thesaurus is to add, as related terms, common everyday expressions corresponding to the formal ones. As another example, consider a firm of solicitors specialising in a certain type of legal case. In such a firm of solicitors it might be relevant to maintain a very detailed thesaurus on certain issues. In KLD this can be done by refining parts of the general thesaurus. In this way the general thesaurus can evolve into a number of case and situation specific thesauri, all within a general framework.

The classification structure

The second expansion of KLD concerns the grouping of the documents. Karnov's Lawbook includes a table of contents organizing the documents into a hierarchy. We would like KLD to include a facility giving an equivalent general view of the structure of the body of laws. With such a facility the user may restrict retrieval to a selected part of the database.

Even though the table of contents in Karnov's Lawbook is carefully produced it suffers from the limitation that every document occurs just once. What we need in KLD is not really a table of contents but rather a classification structure. The difference between the two being that while the former is a hierarchy the latter is some kind of network. The classification structure in KLD is a network arranged in levels. The entire body of laws is divided into a number of legal areas which are further divided into sub-areas. What differentiates this structure from a hierarchy is that documents, and sub-areas, may belong to several legal areas.

One could ask whether the classification structure does anything but duplicate the keyword facility in the thesaurus. The keyword facility could easily be extended to allow the user to restrict retrieval to the documents indexed with terms from a certain branch of the thesaurus. We believe there is a significant difference: the difference between indexing and classification. The purpose of indexing is to indicate that which differentiates one document from the others; the focus is on specificity. Conversely, a classification is meant to group the documents in classes of similar documents; the focus is on similarity (Ingwersen & Wormell, 1990). To a certain extent, the keywords support precision-oriented retrieval (searching with the purpose of retrieving few but central documents) while the classification structure is directed towards recall-oriented retrieval (searching with the purpose of retrieving all relevant and most partially relevant documents).

As with the thesaurus we find it important to allow the user to insert, delete, and update the classification structure. When, in court, a lawyer succeeds in using a law in a new legal area another

lawyer may want to include this law in the legal area in question. The need to modify the classification structure also arises for instance when new legislation grows and interacts with existing laws; for example, laws governing environmental and agricultural issues, relationships between national and supranational legislation. The possibility of modifying the classification structure can also be used to build additional classification structures consisting of just the documents relevant to certain case types, of which the user handles many instances. If such a case-specific classification structure is available, all queries concerning this type of case can be restricted to the documents specifically classified.

Personal notes

The third expansion of KLD was to allow the user to enter new documents and link them to those already in the database. We thought of these new documents as personal notes attached to the legal texts in the same way as the notes made by the editors of Karnov's Lawbook. The personal notes enable writing-while-reading, an important and very practical way of making notes. Notes can be used to make a reference to a textbook, to append the assessment of an expert in the field, etc. A special way of using personal notes is as book marks; for instance, to ease retrieval of the text selections concerning a given case or subject. Book marks somewhat resemble the keyword facility in the thesaurus, but are typically situation bounded.

The personal notes are retrievable in exactly the same way as other documents. Through the personal notes KLD is equipped with limited possibilities to function as a filing as well as a retrieval system. New texts can be imported and placed in personal notes, and any document in KLD can be exported. By expanding the facilities for handling new documents it would be possible to store the users' own documents and other documents internal to the organization together with the body of laws. Our intention is not that each and every document should be included in the IRS but many documents are already in electronic form and some could be included with advantage. There would be one common user interface, information retrieval could cover the body of laws and the internal documents at the same time, and numerous links could be made between legal texts and internal documents.

The final prototype

The prototype was developed in four steps: first the nucleus, then three expansions. This proved convenient as it allowed us to design and implement each facility without thinking about the others. This was in itself a test of the ability of the relational database to expand and change over time. It has been remarkably easy, for instance, to add a new type of document, e.g. the personal notes. Apart from modifying the database, some new code has been written with each expansion. Adding personal notes made it necessary to modify the part of the existing code doing the query processing in order to allow personal notes to be included or excluded from the searches. But expanding the database has not forced us to make any other modifications to the existing code than the ones directly connected to the expansions of functionality.

To allow an evaluation of KLD we will briefly discuss its response time and space requirement. The space requirement is large. The body of text in KLD occupies approximately 4 Mb when stored in a flat file. In the database it occupies almost 19 Mb; i.e. the storage overhead is nearly 400% of the original file size. This is a large overhead, larger than the 50-300% seen in connection with inverted files (Faloutsos, 1985). But we do not find the space requirement prohibitive. We have experimented with 10% of Karnov's Lawbook and have not attempted any kind of compression. Thus the entire lawbook and the necessary working areas would fit into a large hard disk or an optical disk, both rather inexpensive compared to the cost of a complete legal IRS.

To get an idea of the response time we carried out some tests. We made three sets of

queries—simple, medium and complex—by varying the amount of text to be searched, the frequency of the query terms in the body of text, the number of query terms, and the use of proximity operators. The simple and medium queries gave response times of between 1 and 8 seconds (on a Macintosh® IIcx). Most of the complex queries performed similarly. Only three rather extraordinary queries containing 43 query terms gave response times above 10 seconds and the maximum response time was 32 seconds. When judging the response times it should be noted that we did not spend much time optimizing performance. Furthermore, it is important to compare the response time with the amount of time the users find it reasonable to wait. Measures of the response time tend to focus all attention on obtaining minimal response times. We expect that the time the users find it reasonable to wait varies from request to request. Simple requests shall always give rise to short response times; otherwise the IRS becomes a burden hampering the users' work. On the other hand we are convinced that the response time is much less critical when it comes to requests which the user regards as complex. In these cases the quality of the result, the time required to make an equivalent manual search and the like become of equal or greater importance. One of the lawyers to whom we demonstrated the prototype directly expressed that she was willing to accept slower performance if the IRS delivers an answer of high quality. Based on our experience with KLD we find that the hard problem is not to achieve satisfactory response times with relational databases, but to devise ways of achieving high quality answers. This problem does not, however, stem from using a relational database.

On a commercial basis Karnovs Forlag has started working on developing a combined editorial and information retrieval system along the lines presented in this article. Their work involves the entire body of laws, and they achieve response times fully acceptable for practical use.

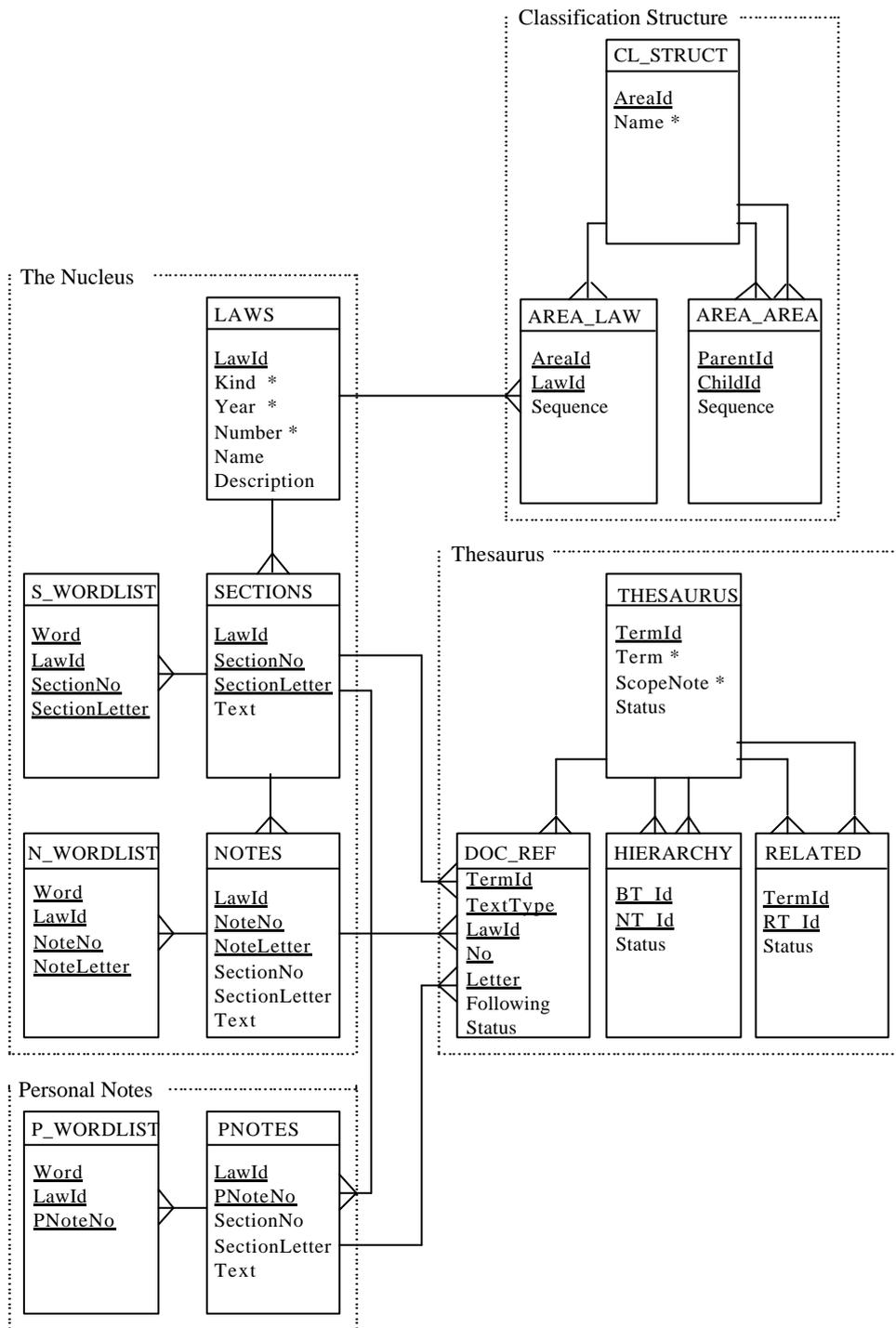


Figure 2 The data model of KLD. The data model satisfies third normal form and consists of a nucleus and three subsequent expansions: the thesaurus, the classification structure, and the personal notes. Primary keys are indicated by underlining, secondary keys by an asterisk.

Document handling and evolution

KLD is an IRS designed to investigate aspects of computer supported legal research. However, we believe that the two major design ideas—to support document handling and continued evolution of the IRS—are of relevance to the majority of IRS for professionals.

Document handling

One of the characteristic features of KLD is the attempt to view information retrieval in its context of people and tasks. We have attempted to integrate KLD with the user's work and with his other computer systems for handling documents, e.g. word processing, electronic mail, and the archival system internal to the organization. The purpose of such an integrated IRS is: (1) To support the user in filing, i.e. collecting, organizing, and storing information for future reference. This is primarily a continuous activity intertwined with the work to be documented, and only to a smaller extent an isolated activity appended to the end of the 'real' work. (2) To support the user in retrieval, i.e. requesting, searching for, and locating documents or supporting references. These documents and references may originate from inside the user's organization or from external sources such as the body of laws in KLD.

Document handling related to documentation work is an essential part of the work of professionals such as lawyers. It is at the heart of litigation and, according to Voges (1988), systems to support this kind of work, so-called litigation support systems, are fairly well known. The office environment is another place where document handling is widespread. Here document handling and systems supporting it are recognized and considered important (see e.g. Morrissey et al., 1986). Within professionals' documentation work and within the office environment, filing and retrieval are intimately related work processes.

On the other hand, in the information retrieval community it is typically assumed that the control, management, and evolution of the IRS and its body of texts are centralised. An essential consequence of this assumption is that the filing activities and the retrieval activities are carried out by different persons. Consequently, it becomes reasonable to assume that an IRS only contains completed documents, i.e. documents which will not be subject to future modifications. This point of view is explicitly stated by Faloutsos & Chan (1988) when they assert that: '*Text databases are traditionally large and have archival nature: there are insertions in them, but almost never deletions and updates.*' Many text databases do have these characteristics; others do not. Using an IRS to support professionals' document handling is one example of an environment in which a system adaptable to changes is required.

To support document handling effectively, an IRS must include facilities making it easy to add new documents and remove obsolete ones. A new law can *de facto* cancel part of an old one without the obsolete part being formally repealed. This happens if the lawmaker does not notice—or want to notice—the conflict. In such cases it is very convenient for a lawyer to be able to add a clarifying note or a new reference. However, addition of new material is not the only kind of modification which must be taken into account. Updates must be considered too. One of the lawyers we talked to maintains a document containing the current fines for a standard type of case in which he is often involved. When the fining practice changes the document must be brought up to date. There is a number of additional ways in which the user may want to modify existing documents. This is the case whether the IRS contains legal material or something else. Morrissey et al. (1986) mention that a doctor may wish to append details of a patient's latest visit to his file, and a user of another IRS may wish to delete a former customer from a mailing list or delete a preliminary version of a recently completed contract.

Evolution

Throughout the design of our prototype a particular aspect of the need for changeability has been in focus: to support evolution. The context surrounding an IRS changes over time. We see a great need for facilities allowing these changes to be smoothly incorporated in the IRS. The need for IRS constructed with a view to evolution arises from the fact that at some point during the development of computer systems it is necessary to freeze the requirement specification. But the context changes continuously, even after the requirement specification has been frozen. Therefore we find it insufficient to deal with change in the systems development process. It is necessary to deal with, and try to incorporate a potential for, evolution in the system itself.

Evolution is a factor which influences manifold aspects and decisions in the design of an IRS. When discussing facilities to support evolution, the people responsible must try to make clear who are the target groups and what kinds of changes are to be considered. At least two target groups can be distinguished: the users and the system developers. As for the kinds of changes, we will restrict the discussion to changes affecting the underlying database; i.e. the changes in the structure of the database, the access to it, and the data in it.

Facilities to support evolution can be directed towards changes in the structure of the database and the access to it. Preparing the IRS for this kind of change is done to ease the work of the system developers responsible for maintaining the IRS. For example, if not already present, the IRS could be extended with information showing when each document was entered, and an online thesaurus could be extended with a relation giving translations between English and French terminology. At construction time it is impossible to foresee all the changes that will be suggested and considered later. What we can do is to strive for general solutions where even structural changes can be smoothly incorporated. This is the major reason we have found it interesting to investigate how a relational database system can give an IRS like KLD the functionality and flexibility needed.

The other major kind of change which should be offered by facilities supporting evolution is changes in the data contents of the database. Changes in the data include the continuous addition of new documents, e.g. the addition of new laws to KLD as they come into force. Furthermore, we believe that it is important to give the user—as an individual or on behalf of a group—certain easy-to-use facilities to alter and adjust the IRS. In KLD this is possible in the thesaurus, in the classification structure, and through the personal notes.

In a study, with students as users, Tague (1981) assessed the feasibility of allowing users to add keywords to documents during their use of an IRS. She encounters some problems, especially in connection with vocabulary and error control, and concludes that post-editing of the user entries is necessary. As post-editing is expensive, Tague finds it unlikely that a facility allowing users to add keywords will be included in an operational system. On the other hand, she acknowledges the idea of permitting the indexing to adapt dynamically to the users' needs. The value of this idea is strongly supported by her retrieval tests where almost 25% of the retrieved documents were retrieved solely on the basis of user-added keywords. There are two reasons why we find it realistic to let the users in our target group enter and delete data: (1) As the users are professionals they have the necessary competence to decide whether a modification is relevant. (2) If the IRS has a reasonable functionality and user interface it will require no knowledge about computers to make the modifications. We stress that allowing the user to modify, for instance, the classification structure does not mean that centrally controlled updates should be dispensed with.

Conclusion

We have carried out a case study consisting of the design and implementation of a prototype of a legal IRS. KLD is only a prototype, but we expect it to be an example of a type of IRS which will become increasingly important and widespread.

The design of KLD has been guided by two major design ideas: (1) To support the user's document handling during documentation work. This involves two closely inter-linked activities, filing and subsequent retrieval. There is more to filing than allowing the user to store new documents; the user must also be given the opportunity to insert the new documents in a classification structure, link them to other documents, attach keywords to them etc. (2) To support evolution. In the development of KLD this led to a dynamic thesaurus, a dynamic classification structure, and personal notes. As a part of the thesaurus, we proposed a facility called reversed indexing, i.e. the use of terms in the thesaurus as keywords with visible references to documents of special interest. In certain situations, reversed indexing seems to offer the user a direct way of getting to the relevant documents.

Karnov's Lawbook has existed since 1924. During its lifetime it has been subject to many changes and extensions; but its overall structure has remained remarkably stable. Similarly, it should be a key consideration to establish computer-based legal IRS within a technical framework stable to a multitude of changes.

KLD was based on a relational database. We expect IRS based on relational databases to offer an attractive combination of efficiency and changeability compared to IRS based on inverted files, at least for small and medium sized systems, i.e. systems with less than 100 Mb of text. Many IRS applications fall into this category. In the information retrieval literature it is common to reject relational databases on the grounds that they are too time and space consuming. Sometimes, this rejection is too simplistic and does not properly take account of the need for the IRS to adapt to changes.

Acknowledgments

We are indebted to Jens Peter Nielsen, manager director of Karnovs Forlag, for making this study possible by giving us access to a machine-readable version of their publication. We also owe thanks to Dorthe la Cour, Peter Ingwersen, and Per Sjøqvist for their engaged participation in interviews and demonstrations. For valuable comments to an earlier version of this article we are indebted to Randi Nygaard Andersen, Liam Bannon, Peter Blume, Hasse Clausen, Olav Green-Pedersen, Jan Hagemann, Martin Jensen, Jens Peter Nielsen, Jacob Nørbjerg, and Otto Ring. Special thanks are due to Claus Topsøe-Jensen, Helge Korsbæk, Erik Kristiansen, and Peter Naur.

References

- BATTY D (1989) Thesaurus construction and maintenance: A survival kit. *Database* **12**(1), 13-20.
- BELKIN N J, ODDY R N, & BROOKS H M (1982) ASK for information retrieval: Part I. Background and theory. *Journal of Documentation* **38**, 61-71.
- BLUME P (1989) *Juridisk informationssøgning* (Legal Information Retrieval), 3. edition. Akademisk Forlag, Denmark. In Danish.
- CODD E F (1970) A Relational Model of Data for Large Shared Data Banks. *Communications of the ACM* **13**, 377-387.
- DATE C J (1986) A Practical Approach to Database Design. In *Relational Databases: Selected Writings* (DATE C J), pp. 417-470. Addison-Wesley, Reading, MA.
- DATE C J (1990) *An Introduction to Database Systems: Volume I*, 5. edition. Addison-Wesley, Reading, MA.
- FALOUTSOS C (1985) Access Methods for Text. *ACM Computing Surveys* **17**, 49-79.
- FALOUTSOS C & CHAN R (1988) Fast Text Access Methods for Optical and Large Magnetic Disks: Designs and Performance Comparison. In *Very Large Data Bases, VLDB: Proceedings of the fourteenth international conference on very large databases*

- (BANCILHON F & DEWITT D J, Eds.), pp. 280-293. Morgan Kaufman, Palo Alto, CA.
- FOSKETT D J (1985) Thesaurus. In *Subject and Information Analysis* (DYM E D, Ed.), pp. 270-316. Marcel Dekker, New York.
- FRØKJÆR E (1989) Controversial Issues of Expert Systems in Public Administration. In *Expert Systems in Public Administration* (SNELLEN I TH M et al., Eds.), pp. 35-54. Elsevier Science Publishers, Amsterdam.
- HERTZUM M & SØES H (1992) *Fuldttekstsøgesystemer til fagfolk* (Full-Text Retrieval Systems for Professionals). DIKU-report 92/10. DIKU, Copenhagen. In Danish.
- INGWERSEN P & WORMELL I (1990) *Informationsformidling i teori og praksis* (Information Mediation in Theory and Practice). Munksgaard, Copenhagen. In Danish.
- KENT W (1983) A simple guide to five normal forms in relational database theory. *Communications of the ACM* **26**, 120-125.
- LEITH P (1986) Fundamental Errors in Legal Logic Programming. *The Computer Journal* **29**, 545-552.
- MORRISSEY J M, HARPER D J, & VAN RIJSBERGEN C J (1986) Interactive querying techniques for an office filing facility. *Information Processing & Management* **22**, 121-134.
- NISHIMOTO H & URA S (1989) Complex view support for a library database system. *Information Processing & Management* **25**, 515-525.
- SOERGEL D (1974) *Indexing Languages and Thesauri: Construction and Maintenance*. Wiley, Los Angeles, CA.
- SPROWL J A (1981) WESTLAW vs LEXIS: computer-assisted legal research comes of age. *Program* **15**, 132-141.
- STRONG G W & DROTT M C (1986) A thesaurus for end-user indexing and retrieval. *Information Processing & Management* **22**, 487-492.
- TAGUE J M (1981) User-responsive subject control in bibliographic retrieval systems. *Information Processing & Management* **17**, 149-159.
- TENOPIR C (1984) Full-Text Databases. *Annual Review of Information Science and Technology* **19**, 215-246.
- VOGES M A (1988) Information Systems and the Law. *Annual Review of Information Science and Technology* **23**, 193-216.
- VON EYBEN W E (1989) Karnovs Lovsamling (Karnov's Lawbook). Updated version of an article in *Danske Opslagsværker* 10-18, 1977. In Danish.