

SISU **dokument**

Survey of Text Retrieval Products

**Peter Rosengren
Marie Bern**

**nr
15**

1. Introduction

Systems for storing and retrieving text documents have been used for many years in specific application areas, such as libraries, governmental institutions and news agencies where the business requires management of large amounts of textual information.

In other sectors the use of text retrieval systems as general tools for information management has so far been limited. But as the amount of electronically produced text documents has increased dramatically during the last two decades, systems for managing large text databases are becoming interesting for most large organisations.

Users are beginning to realise that there are large amounts of company knowledge stored in text documents and are looking for methods to utilise this huge resource [Byte92].

The purpose of this document is to give a short introduction to text retrieval and to give some examples of text retrieval packages available on the market.

The document is organised as follows. In chapter 2 we will give a short introduction to text retrieval. In the following chapters we survey some text retrieval products. The survey is by no means exhaustive. Today, there exists at least 30 text retrieval products of which many are PC-based. The reader that wants a more complete overview of the text retrieval market is referred to the book "Text Retrieval - a directory of software" which is published every three years and provides a detailed market survey of text retrieval products [Kimberly90].

The first three products presented, PC-Search, ZyIndex and SearchCity are examples of what could be the choice for a small to medium sized installation. The typical usage would be to use the products to search for documents on a single PC or in a PC-network. The next two, BasisPlus and Topic, are examples of what could be the choice for a large scale installation with thousands of users. The last product, Fulcrum Ful/Text is targeted towards application developers that want to incorporate text retrieval functionality in their applications.

This survey has been produced within the Intuitive project [Rosengren93].

2. Characteristics of Text Retrieval Systems

Text retrieval products can be divided into two categories, sometimes referred to as *finders* and *keepers*.

Finders can search for documents on a hard disc, server, CD-ROM etc. but has no functionality for storing and managing text documents.

A *keeper* has search functionality but also uses a database to store the documents. Some products use standard relational database technology while others use their own proprietary database. The database functionality offers advantages for maintaining the documents, i.e. it is possible to define access rights, security control etc. The drawback is that the documents must be converted into the database format which is usually ASCII-format. On the other hand the database always keeps a pointer to the original text.

The general trend among text retrieval developers is to adopt a client/server architecture. Another trend is that text retrieval products often are being integrated in other products such as groupware and work flow management systems.

Logically a text retrieval system consists of two modules - an *indexing engine* and a *search engine*. However, in many products these two functions are integrated into one. For the sake of clarity we will continue the discussion as if there are two separate modules.

2.1. Indexing

The indexing engine prepares the document space for searching. The index engine does so by scanning through each document and extract all the words in the documents. The words are then put in an index where each occurrence of the word (i.e. the document and the position within the document) is stored. It is through the index that it is possible to search for documents.

To index all words in a document is referred to as *full-text indexing* and it is the most common indexing technique used in commercial packages. Older systems relied on *keyword indexing*, where keywords were attached to the document or to specific fields in a document at design time.

One drawback of full-text indexing is that the index might grow as large as the document space itself. Another drawback is that if the original documents are changed this is not reflected in the index file. However, some of the systems discussed in this document at least warn the user if there are inconsistencies between the index and the documents.

To avoid a huge amount of meaningless search terms, such as "a", "an", "is", "it", "then" etc. *stopword lists* are used. The stopword list contains all words to be

excluded from the indexing process. Normally the stopword list can be edited by the user.

Naturally an important aspect of the indexing engine is which document formats it can read without any pre-conversion. Text retrieval packages vary in this aspect and the reader is recommended to check the document formats handled when evaluating a text retrieval package.

An alternative to full-text indexing is to simply avoid indexing and instead scan through the each documents for every user query. This is referred to as *free-text scan* and of course requires very powerful hardware.

2.2. Searching

The functions of the search engine can be grouped into two - the search language and the search support functions.

2.2.1. Search Language

The *search language* is used by the users to specify their queries. The search language typically offers the following retrieval facilities:

- *Boolean search*, which allows a user to specify a boolean expression of terms, for example "(multimedia AND databases AND 3-schema) OR (MMDBMS and 3-schema)". Normally the operators AND, OR and NOT are available. Nesting of expressions is normally supported.
- *Field search*. Often it is possible to specify fields within a document, for instance author, title, version, date et c. The document fields can be referred to in a query, i.e. it is possible to restrict the search to certain fields of the documents. One example is to search for all documents with the words "multimedia" and "databases" in their titles.
- *Proximity search*, which allows the user to specify the context in which a term occurs. One example is to specify that a word should occur within a number of words from another word, for instance "databases WITHIN 10 multimedia" retrieves documents where the word "databases" occurs ten or less words from the word multimedia. Normally proximity could also be specified by sentence or paragraph distances.
- *Phrase searching*. This means that a whole sequence of words, i.e. a phrase, can be entered as a search criteria. If this feature is well implemented the system should be able to ignore any stopwords used in the phrase. Unfortunately this is not true for all commercial products, which sometimes leads to no hits because a stopword like "the" was used in the phrase.

- *Range Searching* allows a user to specify an interval as a search criteria. An example is to search for all numerical values between 1 and 10. Some products also offers range searching for character strings. An example is "Peter..Roland" which would retrieve all documents containing a word in the interval "Peter..Roland".
- *Wild Card Searching* means that wild cards such as "*" and "?" can be used to specify any sequence of characters or any character. Examples are "Pe*", which would match all words beginning with "Pe" and "I?M" which would match all three letter words beginning with "I" and ending with "M".

2.2.2. Search Support

The role of *Search Support Functions* is to assist the user in the search process. Typically examples of what is offered are:

- *Search refinement*, which allows iterative searching. A user can choose to narrow the search by adding more search terms or broadening the searching by using fewer search terms.
- *Index Browsing* allows a user to browse through all the terms in the index and select search terms directly from the index list. One advantage of this is that the user can be sure of only using search terms that actually exist in the documents.
- *Synonym Lists* makes it possible to specify synonyms. When the search is conducted all synonyms for a particular search term is considered to be equal to the search term.
- *Thesaurus* which is more sophisticated variant of synonym lists. A thesaurus can contain synonyms, but also plural and possessive forms, word variants et c.
- *Concept Definitions* provide a higher level of abstraction than words. Normally a concept can be defined as any query. For instance it is possible to define "multimedia databases" as the following boolean expression:

"(multimedia AND databases) OR (MMDBMS) OR (multimedia AND DBMS)"

"Graphical user interface" can be defined as:

"GUI OR (graphical AND user AND interface) OR (graphical AND UI)".

Once the definitions have been set up they can be used as any search terms. However, the concept functions in various products differ in sophistication.

2.2.3. Advanced search facilities

Some advanced systems also offers *relevance ranking*, *fuzzy searching* and *topic trees*. A system that supports relevance ranking presents the user with a ranking list based on how well the documents match the query, i.e. a relevance status value is calculated and assigned to each document. Instead of telling the user whether a document meets the criteria specified in a boolean expression or not, a relevance ranking system tells the user that a document to a certain degree (for instance 0.75) is relevant to the query.

Fuzzy searching means that the search engine can locate documents with words close to the specified, for instance if a user specifies "peter" as a search term documents containing words like "peeter" or "petter" will also be retrieved.

Topic trees allows a user or a system administrator to define to which extent different terms relate to each other in a weighted tree.

Systems offering fuzzy searching and concept trees also presents the user with a relevance ranking of the documents retrieved.

2.3. Presentation

A third aspect of a text retrieval system is how the result of a search is presented to the user. Here are some examples of what can be expected from a text retrieval system.

- *Browsing list*. Most systems present the user with a list of the names of the documents that matched the query. This list can be scrolled and any document can be viewed in at least ASCII format.
- *Number of hits per document*. It is common to show the user the number of hits within a document.
- *Hits in context*. A user-friendly feature is if the hits are displayed in their context, for instance if the sentence before and after the hit is shown. This gives the user a possibility to evaluate whether the hit is relevant or not.
- *Hit indication in document*. When a user opens a document the hits within the document are highlighted. Browsing through hits within a document and between documents is normally supported.
- *Viewing of native document*. A nice feature that is available in some text retrieval packages is the possibility to view a document in its native format, or to launch the word processor that produced the document.

3. PC-Search

PC-Search is a full-text indexing and retrieval system for Windows 3. PC-Search is built upon the Fulcrum Ful/Text engine, see chapter 8, but is sold and distributed by an independent company called Intunix in Switzerland.

PC-Search consists of an administrator's module and an end user module. The administrator module allows a system administrator to define *collections* of documents to be searched and retrieved. The standard version of PC-Search handles MS Word, WordPerfect and ASCII file format. Filters for other formats are available as add-on products. The figure below shows how a collection is defined.

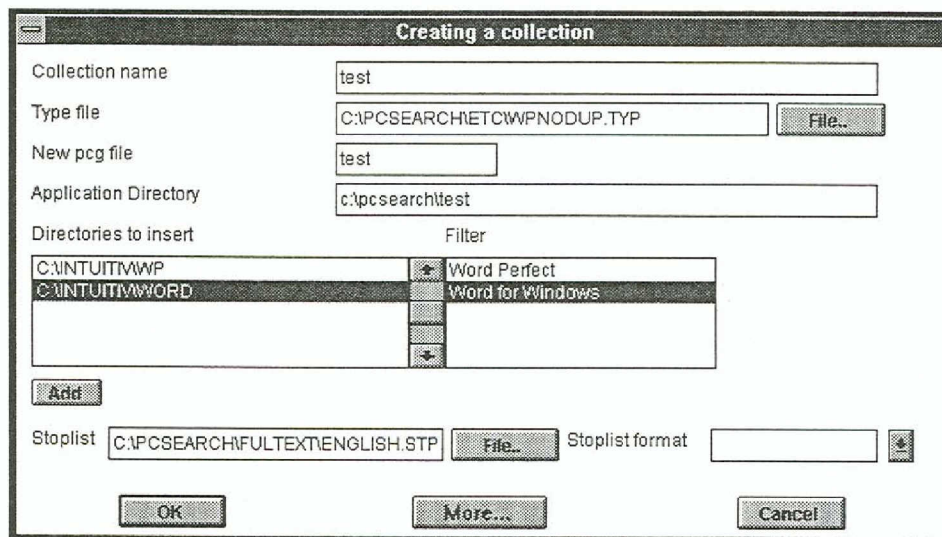


Figure 3.1 The user interface of the system administrator's module in PC-Search.

A default stoplist file is provided and can be edited. Once the collection is defined, indexing takes place automatically.

The end-user module allows a combination of free text search and field search. One of the strengths of PC-Search is that it can directly use the field definitions of Word and WordPerfect. This allows a user to reuse information that has already been added to the documents, see the figure below

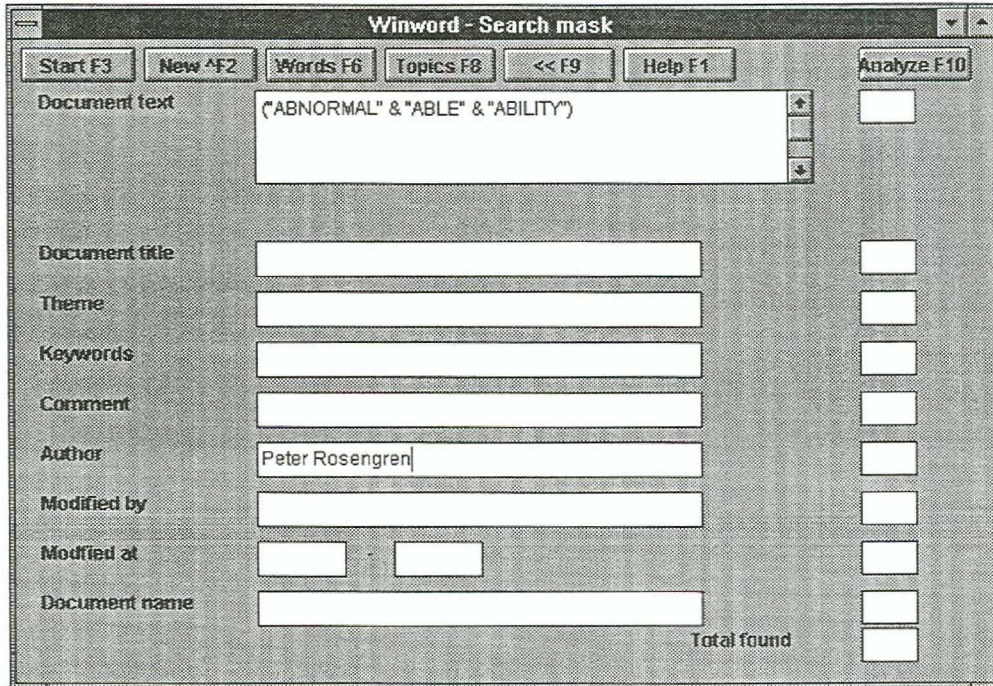


Figure 3.2 The end user's search interface of PC-Search.

The query language allows specification of boolean, phrase and proximity searches. Index browsing and direct selection of index terms as search terms is possible, see figure below:

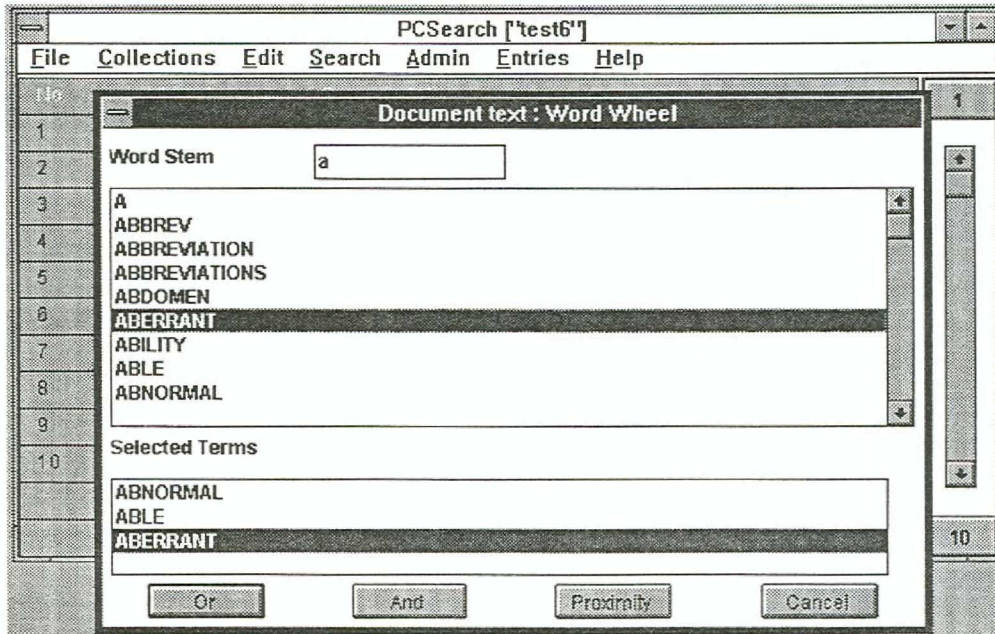


Figure 3.3 PC-Search allows direct selection of index terms as search terms.

Once the documents are retrieved, a result list is presented to the user. The list is sorted according to number of hits per document, see figure below.

No.	modified at	Document title
1	18. June 1993	Common Technical Deliverable v.1.0
2	17. March 1993	
3	30. April 1993	DDA2DRA.WP
4	6. May 1993	MI1-04
5	27. April 1993	MI1-FTR
6	1. March 1993	
7	1. March 1993	
8	14. June 1993	PPR12.WPM
9	30. April 1993	DDA32DRA.WP
10	4. March 1993	

Figure 3.4 The result of a search as displayed in PC-Search.

When the user double clicks on an item in the result list, the corresponding document is displayed. The hits within the document are displayed as well, see figure below.

Document viewer 2 of 10

No.	hits
1	18
2	30
3	30
4	6
5	27
6	1
7	1
8	14
9	30
10	4

Conceptual Query Language
Version 0.1
Peter Rosengren
Swedish Institute for Systems Development
Document WPIT.T2

Conceptual Query Language
INTUITIVE P6593
Abstract
This document is a first initial draft specification of **SQL**, the internal query language to be used within Intuitive. It discusses the requirements for **SQL** and gives examples of how **SQL** can be used. **SQL** is designed to support both conceptual querying and content querying. A brief discussion about how **SQL** can be translated into SQL is also presented.

Identifier WPIT-**SQL**-wp-0.1-SISU-PR

Doc - F7 Hit - F5 Print Options TextInfo
Doc - ^F7 Hit - ^F5 Copy Search Modify

Figure 3.5 A document and the hits within the document are displayed to the user.

4. Zyindex

Zyindex is a document retrieval system that runs under Windows 3. Zyindex consists of two modules - *ZyBuild* and *ZyFind*.

The ZyBuild module is used to define the document database and to construct the index. ZyBuild is straightforward to use and an index can be set up within a few minutes. The only thing the system administrator has to do is to define which directories and file types are to be included in the index, see figure below.

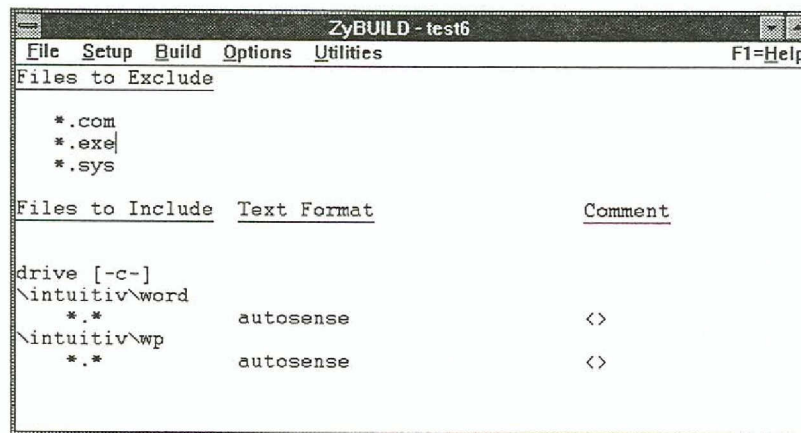


Figure 4.1 The system administrator's module of ZyIndex.

The search interface, ZyFind, is shown below. It shows how a user can select search terms directly from the index. Search terms could also be typed in.

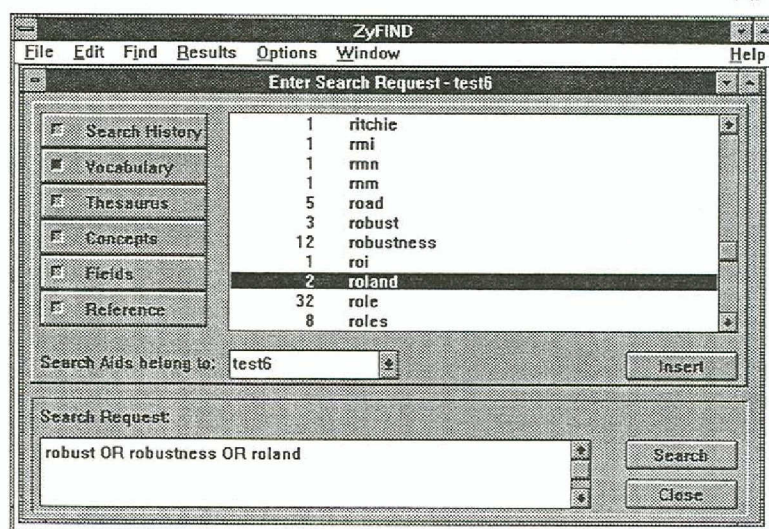


Figure 4.2 The user interface of ZyIndex.

The search language of ZyIndex is quite sophisticated. It allows construction of queries containing Boolean operators, wild cards, parentheses, proximity and many other restricting constructs. A query could for example look like:

Intuitive AND >=2 OF [Sisu, Cap Gemini, City, Ibermatica]
 IN AUTHOR {Rosengren} AND IN TITLE {Survey}

The result list of documents is displayed with the number of hits per documents indicated.

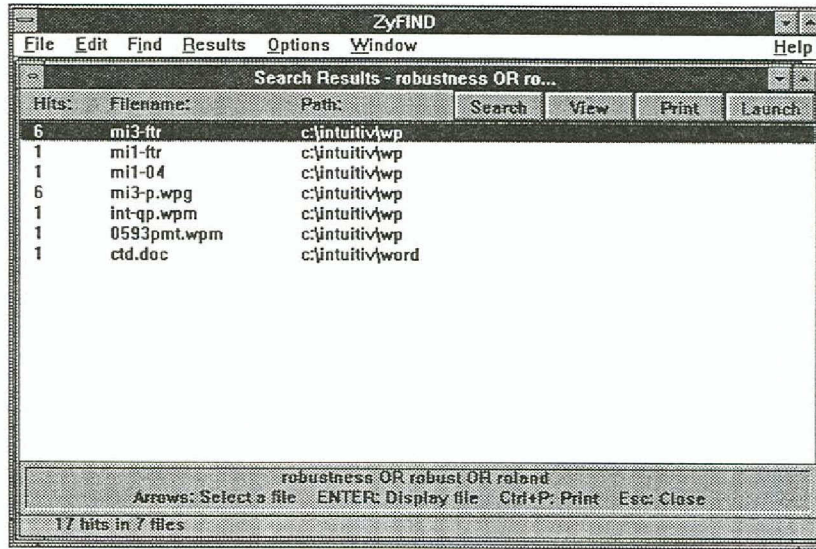


Figure 4.3 The result of a search is displayed.

As in PC-Search the hits within a document are highlighted, see figure below.

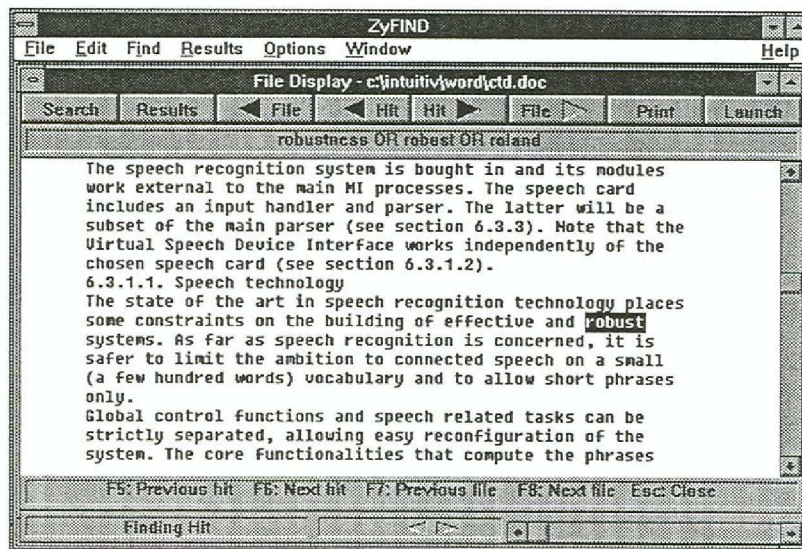
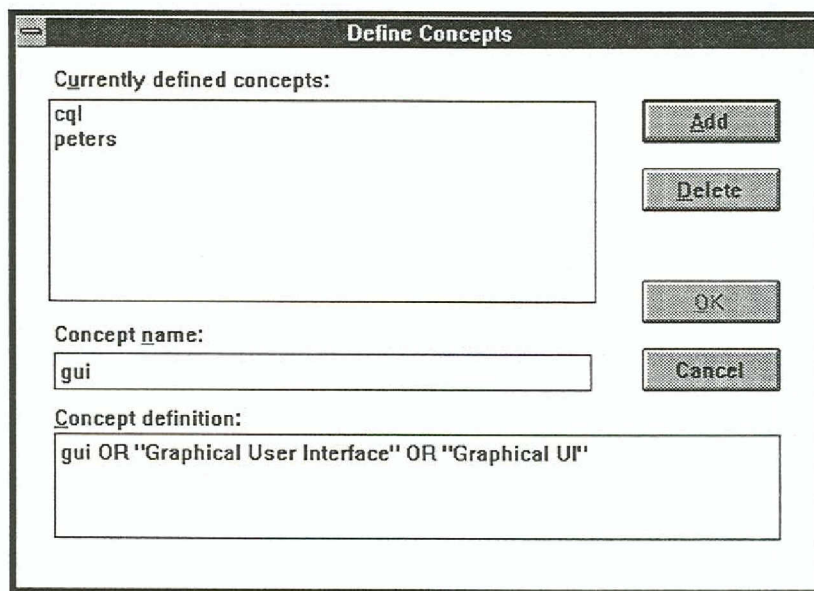


Figure 4.4 It is possible to browse through the different hits in a document.

A user can browse through the results either hit-by-hit or document-by-document. This is done by using the navigation arrows, see figure above.

Other useful features are:

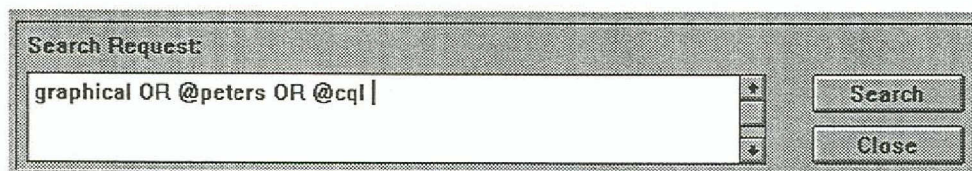
- The use of *synonyms*, that are stored in a user modifiable thesaurus.
- The definition of *concepts* or topics. A concept is defined by clustering words together. The concept PAY-TV could for example be defined by: "PAY-TV = Cinemax OR Movie channel". See figure below for the concept definition dialogue box.



The image shows a dialog box titled "Define Concepts". It has a section "Currently defined concepts:" with a list box containing "cql" and "peters". To the right of this list are "Add" and "Delete" buttons. Below the list box is a "Concept name:" label and a text input field containing "gui". To the right of this field are "OK" and "Cancel" buttons. Below the name field is a "Concept definition:" label and a larger text input field containing "gui OR 'Graphical User Interface' OR 'Graphical UI'".

Figure 4.5 The concept definition box.

Once defined, a concept can be incorporated in any search, see figure below.



The image shows a search request dialog box. It has a label "Search Request:" and a text input field containing "graphical OR @peters OR @cql |". To the right of the input field are "Search" and "Close" buttons.

Figure 4.6 Concepts can easily be incorporated in any search.

- The *progressive search* that allows a user to narrow a search result by constraining the next search to the results of the last one.

Another feature of ZyIndex is that hits can be displayed within their context, see figure below:

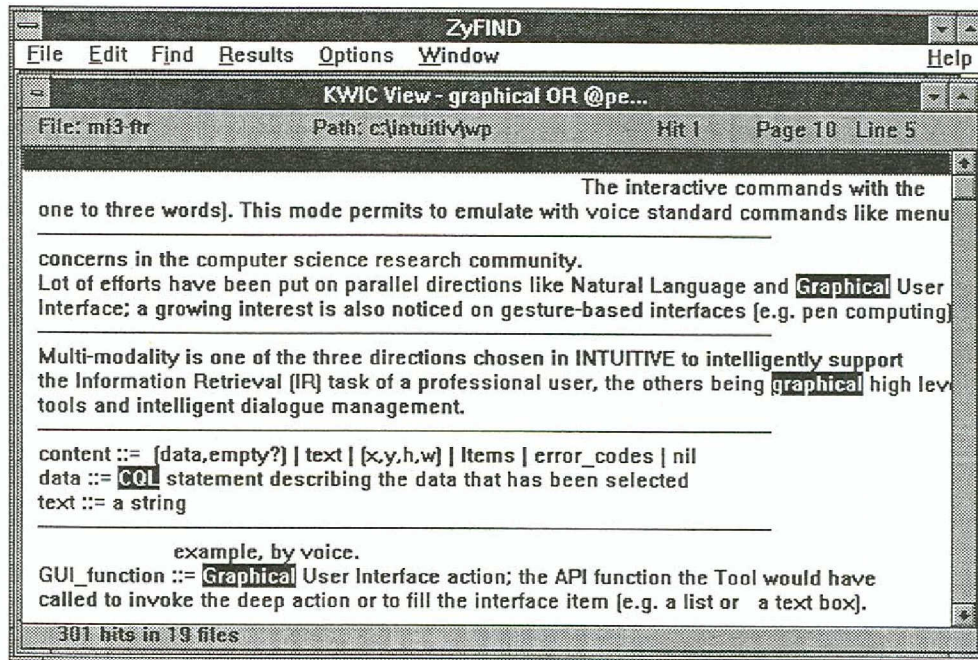


Figure 4.7 Hits are displayed in their context.

A user can define the number of rows above and beneath the hit that should be displayed.

Zyindex further has support for hyperlinking. This means that links can be established from text documents to other documents, for example text, graphics, e-mail or database files. Besides linking to other document, notes can be attached to text and image documents. Attaching a note to a document does not modify the original document. It is also possible to search in notes.

The most important word processors and text formats supported are WordPerfect, Word, MS Write, WordStar, ASCII and ANSI.

An application program interface (API), called Zyindex Developer's Tool Kit, is available for Zyindex. The API is accessible from C. The tool kit enables developers to embed Zyindex document retrieval technology into their own application products.

5. SearchCity

SearchCity is a text retrieval program from the Chilean software house ARS Innovandi. The user interface of SearchCity is well-designed and modern compared with many other text retrieval packages. The user is offered boolean search, enhanced with proximity search as well as phrase searching. After a query the user is informed of how many documents matched the query and how many hits there are in these documents. See figure below for an example of a user query in SearchCity.

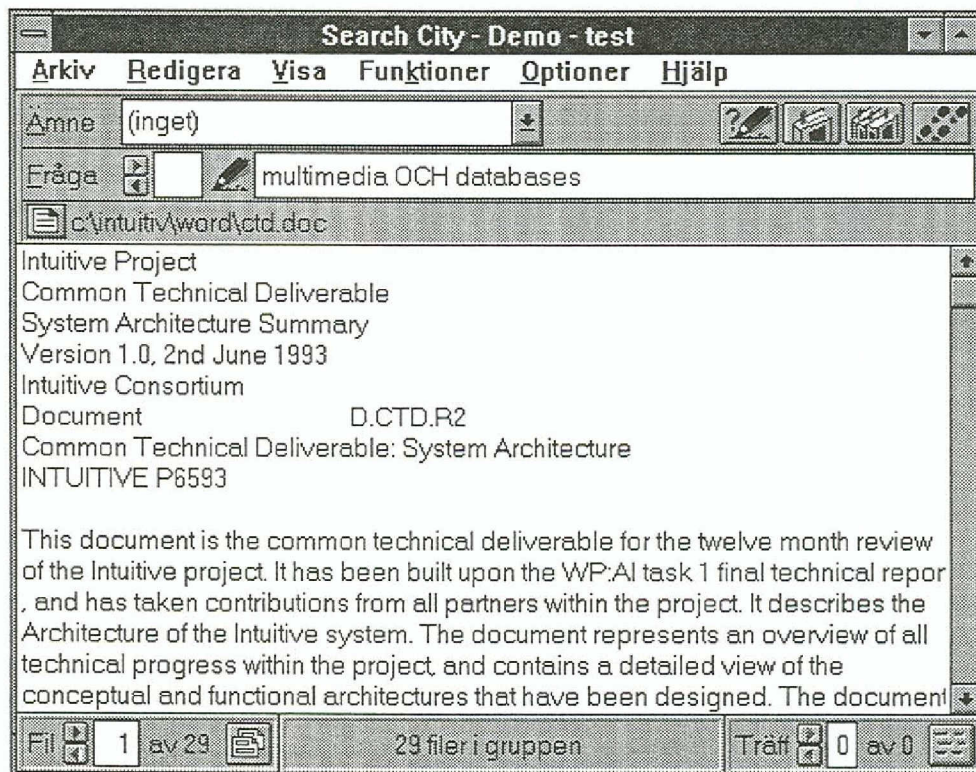


Figure 5.1 A user query and a retrieved document. Note that the number of documents matching the query is displayed as well as the number of hits within the documents.

One limitation of SearchCity is that it is not possible to display the original file or launch the corresponding word processor.

Unfortunately, SearchCity does not offer direct selection of index terms when querying. A query can be saved as a concept and reused later. However, the implementation of concepts is not as sophisticated as in for instance Topic or ZyIndex. A limitation is that only one concept can be used per query.

Like most text retrieval packages SearchCity supports synonym lists that can be edited by the user.

When hits are displayed, SearchCity shows each hit and the context of the hit in a dialogue box, see figure below.

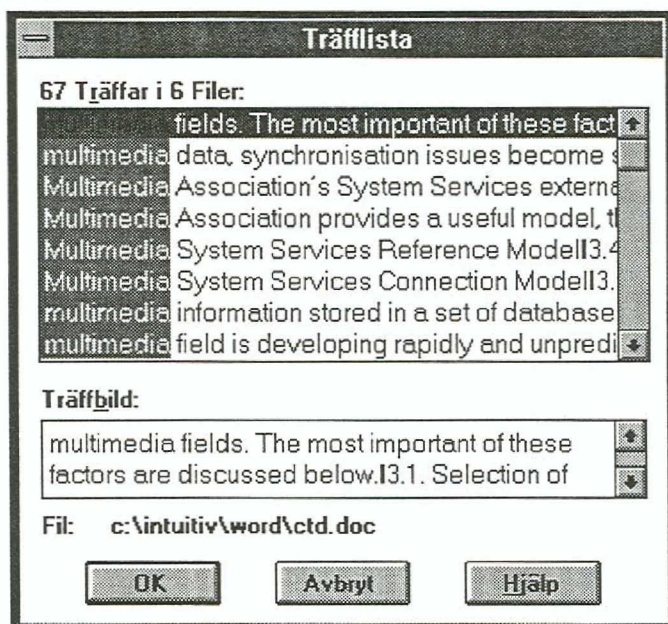


Figure 5.2 Hits and the context of the hits are displayed in a dialogue box.

SearchCity keeps track of the different queries a user poses and offers a search history list where previous queries immediately can be accessed, see figure below.

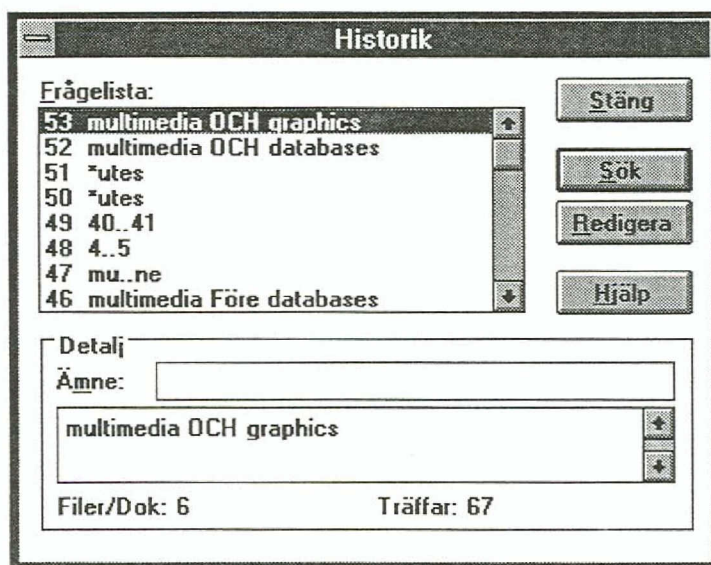


Figure 5.3 A search history list.

Construction of an index is straightforward with SearchCity. It only requires that the user chooses the directories to be included and which stopword list, synonym file and character filter to be used.

SearchCity accepts Word, WordPerfect and ASCII files. The indexing engine is fast compared with PC-Search and ZyIndex. A nice feature is that SearchCity warns the user if the original files have been changed.

6. Basis Plus

BasisPlus is an example of a text retrieval system that does more than only retrieve documents. It also stores and manages huge amounts of documents. The strengths of Basis Plus are its own relational database and its support for SGML (Standard Generalised Mark-up Language). BasisPlus has been chosen as the document database for the US Defence project JCALS.

BasisPlus actually consists of four different products

- Basis Desktop
- BasisPlus
- Basis SGML Server
- TechLib Plus

Basis Desktop is the client software that provides the user interface, see figure below.

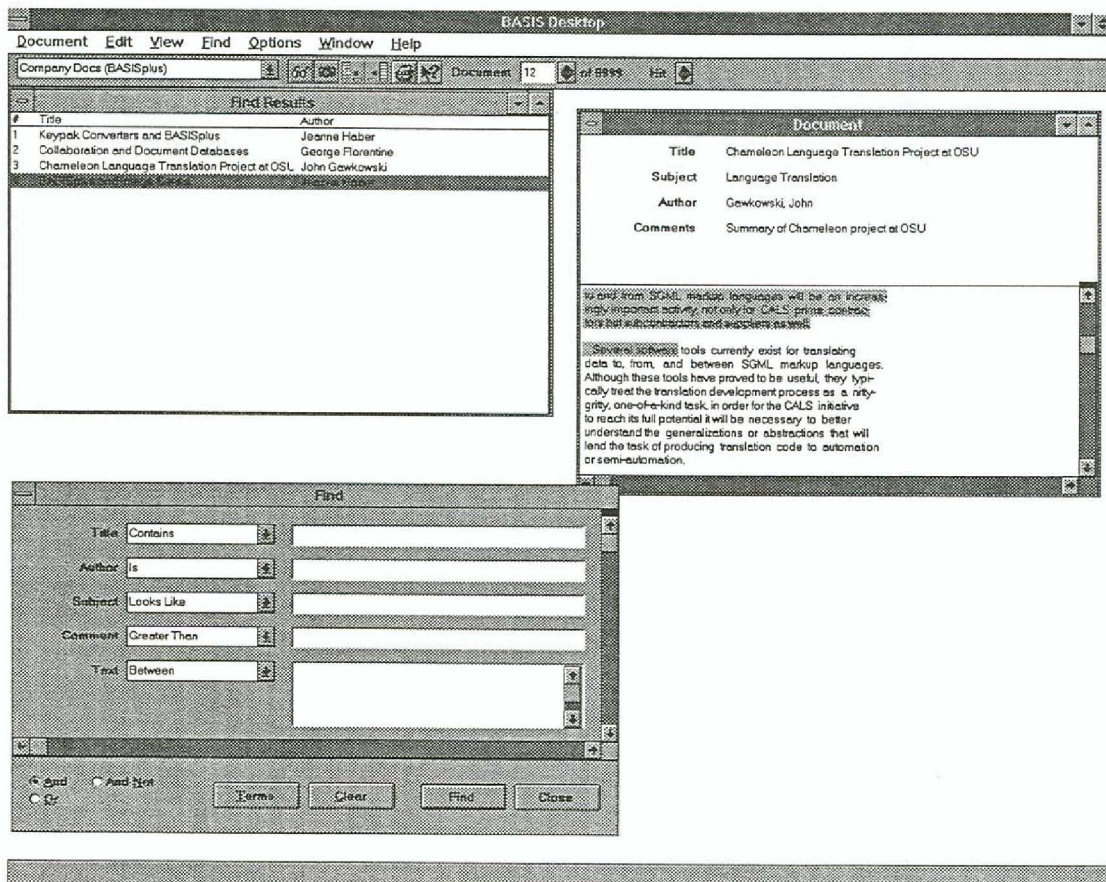


Figure 6.1 The user interface of Basis Desktop.

Basis Desktop contains most of the features mentioned in chapter 2.

BasisPlus is the text database manager that stores and manages the document collections.

Basis SGML Server makes it possible to store and retrieve SGML documents. An SGML-tagged document is parsed and the text is divided into different components depending on the SGML encoding. The individual components are stored in a relational database. In this way it is possible to search through titles, abstracts, headings, appendices etc.

Individual components can also easily be re-used when new documents are produced.

TechLib Plus is a customised application for library use.

There are approximately 2600 installations of BasisPlus world-wide, among which approximately 60 percent are Vax/VMS-based.

BasisPlus was originally developed at the US research institute Batelle Institute.

7. Topic

Topic is a text retrieval system from Verity. Like BasisPlus it is intended for large scale installations. It is one of the most advanced text retrieval packages available. One of its major strengths is that it works with relevance ranking. The result of a query is a sorted list of documents each assigned a score calculated by the system, see the figure below where documents have been assigned a relevance ranking.

The screenshot shows the Topic software interface. At the top, there is a menu bar with options: File, Edit, View, Query, Navigate, Launch, Window, and F1=Help. Below the menu bar, there is a text input field containing the query: "toxicity,sugar,trail". Below the input field, there are two buttons: "Retrieve" and "Merge". Below the buttons, there is a table showing the results of the search. The table has three columns: "Score", "documentnumber", and "title". The results are sorted by score in descending order.

Score	documentnumber	title
0.75	87030007	TOXICITY OF TOPICAL SUGAR
0.50	86030007	ULCERES DE JAMBE. ULCERE VEINEUX NON COMPLIQU@
0.50	85030020	SUGAR AND WOUND HEALING (LETTER)
0.50	85030010	USE OF GRANULATED SUGAR IN TREATMENT OF OPEN MEDIA
0.50	81030039	PRESSURE SORES. PATHOGENESIS, PROPHYLAXIS, AND TRE
0.50	81030010	DIE BEHANDLUNG CHRONISCHER WUNDEN
0.50	79030070	/HISTOLOGICAL EXAMINATION IN CHRONIC TOXICITY TEST OF
0.50	79030069	/CHRONIC TOXICITY OF DEBRISAN/
0.50	79030068	/HISTOLOGICAL EXAMINATION IN SUBACUTE TOXICITY TEST O
0.50	79030067	/SUBACUTE TOXICITY TEST OF DEBRISAN IN RATS/
0.50	79030066	/ACUTE TOXICITY TEST OF DEBRISAN/
0.50	70030003	RADICAL OPERATION FOR CARCINOMA OF THE VULVA. A NEW
0.49	86030015	THE USE OF SUPER - SATURATED SUCROSE SOLUTION FOR C
0.49	84030015	DEXTRANOMER IN CHRONIC WOUND HEALING
0.49	79030002	DECUBITUS ULCERS: A COMPARATIVE STUDY

Figure 7.1 Topic presents the result based on a relevance ranking of the documents.

Topic uses so called *Concept Retrieval*. This means that a user can define *topics*. This is done by constructing topic trees. In a topic tree a user defines other topics that is related to the first topic. It is also possible to assign relevance weights to the relations between two topics, see figure below for an example.

Once a topic tree has been constructed it can be saved and reused. The relevance weights in the topic tree can be changed. For instance if a user would like to search for "multimedia" but focus on the use of multimedia in homes he would increase the relevance weight for "Home Computing" to maybe 0.9. Besides concept search Topic allows Boolean and word searches.

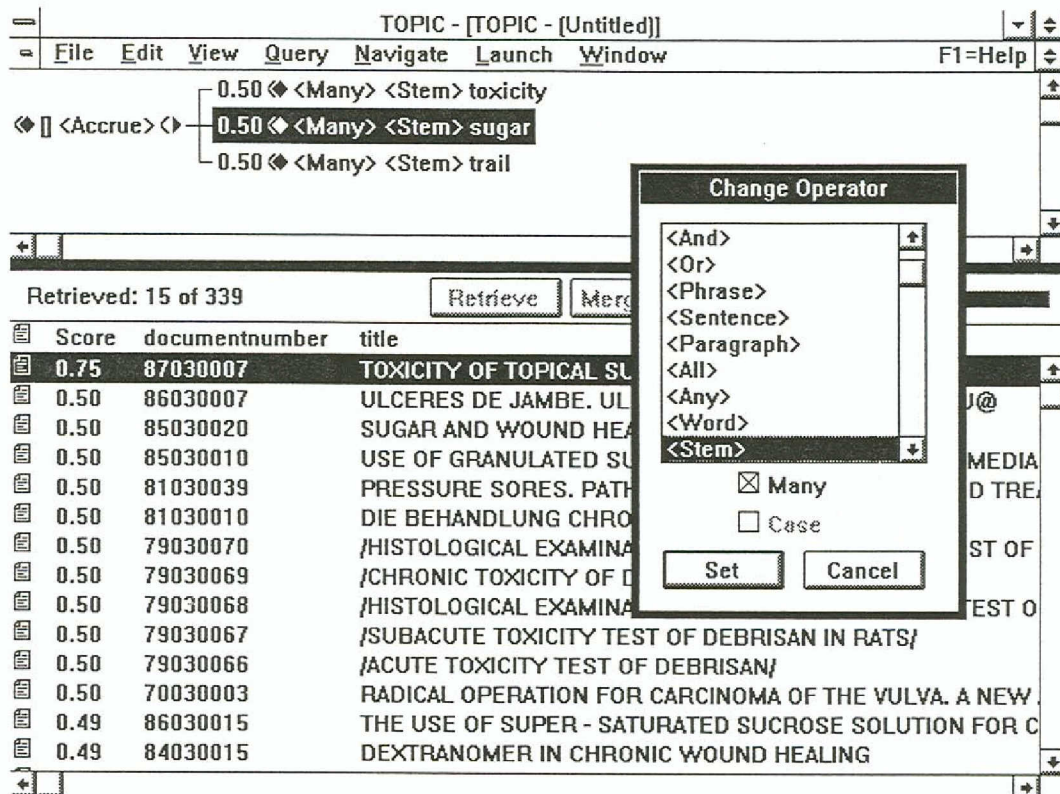


Figure 7.2 Topic trees can be defined by a user or a system's administrator.

Topic manages documents stored in different formats - WordPerfect, MS Word, Excel, Interleaf, SGML. If a Word document is retrieved Topic can activate MS Word to display the document.

Like ZyIndex, Topic supports hypertext links which make it possible to annotate a text document with an image or a note. The linked object is displayed when the document is retrieved.

There is a link to SQL databases called Topic SQL Bridge. The SQL databases supported are Oracle, Sybase, Ingres and Informix.

Topic runs on PC, Mac, OS/2, UNIX and VMS.

Verity has been successful in signing agreements with other product vendors to bundle Topic as a search engine. One example is Lotus Notes which uses Topic as a search engine.

A product for filtering real-time information exists and it is called Topic Real-Time. This is for instance used by the Swedish Interpol to filter out urgent messages coming from other Interpol organisations in Europe. A user specific search profile can be defined. Topic Real-Time reads through incoming X400-messages and if there is a hit the message is automatically sent to user that has the search profile.

The most notable installation of Topic is at CIA in USA which have among 9000 Topic users.

8. Fulcrum Ful/Text

Fulcrum Ful/Text is a full-text indexing and retrieval software library that allows system developers to incorporate full-text retrieval capabilities into their applications. The retrieval facilities are accessed using the "C" programming language.

The text documents are stored in *collections*. A document collection is usually defined by a set of dictionaries containing the document files. Each collection could contain documents of various formats. To overcome differences in text format, Ful/Text uses filters which do not alter the documents themselves.

The Ful/Text API contains support for developing custom document filters if needed. Ful/Text indexes every document in the collections.

Searches can be conducted with a wide range of query parameters. Beside words, the query can comprise phrases, wild card characters and Boolean operators (AND, OR, NOT). Numeric range, date-delimited and proximity searching are supported.

Specific weights can be assigned to search terms to specify their relative importance. Suppose the word "cat" occurs two times in a document and "dog" three times. With weight 1 assigned to both "cat" and "dog" the document would be given the rank 5. If the weight 10 is assigned to "cat" and the weight 5 is assigned to "dog" the document would be given the rank 35.

Ful/Text supports user-defined thesaurus and word variant rules. These features allow the search terms to be expanded into a number of equivalent search terms.

Another interesting feature of Ful/Text is *Intuitive searching*. When a user has retrieved a document it is possible to use this document as the basis for another search of similar documents. The user marks a section of the document or uses the whole document as input for the new search. Ful/Text then generates a set of weighted terms, a so called *document vector*, which is used to search for similar documents.

Fulcrum has been very successful in the CD-publishing industry and Ful/Text is often bundled in CD-applications. One example is CD Answer from Sun Microsystems where Ful/Text is used as the search engine.

Fulcrum supports a client/server architecture. The Ful/Text server is available for OS/2 and Unix, while Ful/Text clients are available for DOS, MS Windows, Apple Macintosh, OS/2 and Unix. Fulcrum has a strong position on the Unix market with a 70 percent market share.

The Fulcrum family consists of three other tools - Fulcrum SearchServer, Fulcrum SearchSQL and Fulcrum SearchTools. The Fulcrum SearchServer incorporates Ful/Text technology and offers an API consisting of 26 calls. The API is based on the SQL Access Group's Call Level Interface (CLI). Fulcrum SearchSQL is the query language for communicating with the Fulcrum SearchServer and it is

based on SQL (Structured Query Language). Finally, Fulcrum SearchTools is an application developer's kit which is available for Visual Basic and Windows developers as for C developers working in OS/2 or Unix.

9. Other systems

As was mentioned in the beginning there are at least 30 text retrieval products available. It is not possible to cover all of them in this survey. We will here briefly mention some other products.

Spirit of Systex is a french text database system that allows queries in natural language. The system analyses the natural language query and computes the semantic proximity between the query and the documents in the database.

The documents are stored as Spirit documents and is pre-processed by linguistic text processing methods. This makes it possible to detect misprints. The syntactic analysis can further recognise and identify homographs, idioms and compounds. Every word is given an informational weight that is based on the frequency of the word in the database.

Spirit of Systex has been developed by three French information retrieval researcher and its major market seems to be in France.

TRIP, is well-known in Sweden, and there are many installations at governmental institutions. Its major platform is Vax-machines.

BRS/Search is one of the old-timers in text retrieval. Its main emphasis has been its on-line database subscription services.

PixTex from Excalibur is known for its neural network algorithms to develop pattern indices of textual data, which is particularly useful if the document has gone through an OCR process.

10. Product Listing

Product	Comments	Vendor
Basis Plus	SGML support, built in relational database, client/server, API	Information Dimensions Inc., USA.
BRS/Search	On-line databases	BRS Information Technology, USA.
Fulcrum/Fultext	API, CD-publishing, SQL support, client server.	Fulcrum Technologies Inc., Canada.
PC-Search	Built with Ful/Text, client server, index browsing, Word and WordPerfect search masks.	Intunix AG, Switzerland
PixTex	Neural network, fuzzy searching	Excalibur, USA
Search City	hits in context, well designed user interface	ARS Innovandi, Chile.
Spirit of Systex	Natural language query, informational weights to word.	Systex, France.
Topic	Concept retrieval, hyperlinks, SQL support	Verity, USA
TRIP	Swedish, Vax-machines	TT-Trading, Sweden
Zyindex	advanced query facilities, hits in context, index browsing.	Information Dimensions Inc., USA.

11. References

- [Byte92] Byte, "Managing Infoglut", July 1992, special Byte-issue about text retrieval.
- [Kimberley90] R. Kimberley, "Text Retrieval - a directory of software", 3rd edition, 1990, Gower Publishing Ltd, England.
- [Rosengren93] P. Rosengren, U. Wingstedt, P. Kool, M. Bern, "Accessing Information in Large Corporate Databases - The Intuitive Approach", SISU Report 1993:03.

SVENSKA INSTITUTET FÖR SYSTEMUTVECKLING

SISU

Electrum 212, 164 40 Kista
Isafjordsgatan 26
Telefon 08-752 16 00 Telefax 08-752 68 00