CD-ROM Chemical Databases: The Influence of Data Structure and Graphical User Interfaces on Information Access

Martin G. Hicks

Beilstein Institute, Varrentappstrasse 40-42, 60486 Frankfurt, FRG

Received June 30, 1993*

Access to chemical information is determined by not only the performance of the retrieval system but also the functionality of the graphical user interface. A CD-ROM based system has to contend with the inherently long disk access times, which are shown to be able to be overcome with a combination of optimization of the data structure and multiple storage techniques. The inflexibility of present day indexing systems that do not provide an easy link between the mental image of a chemical concept, the iconic representation, and the standard query languages, rely on a highly functional user interface to make the transition as easy as possible. The ideal access, as defined by the acronym WYTIWYG (*w*hat *y*ou *t*hink *i*s *w*hat *y*ou *g*et) is still in the future. A user interface has been developed for Current Facts which allows easy information access by giving the user tools to help with the common search classes of feedback searching, navigation and browsing, index browsing, and linked browsing. These functions often implemented as hyperlinks, which are defined in terms of standard, branched, interrupted, and intelligent hyperlinks.

INTRODUCTION

Drowning in Data-Starved of Information. Chemistry is one of the best documented sciences; the data although well defined and well ordered are nevertheless heterogeneous. Thus, while there are, in theory, always structure/property relationships for all compounds, in practice, the property information is, on average, unknown for a particular compound. The amount of data is increasing at a fast rate, thus making use of the data—being able to turn it into valuable information—is a central theme when designing a database product. The problem needs to be addressed in two ways. Firstly from the side of the system performance; systems need to be devised which can retrieve the answer set for any required query in an acceptable amount of time. Furthermore, user interfaces need to be developed which allow the user easy access to the data. The information is not stored in a format that allows the chemist to arrive at his target without specialist knowledge. It is the role of the user interface to act as intermediary, to provide functions, which allow this to be possible. The advent of graphical user interfaces, now practically the standard type of user interface for commercial software, have made this easier to achieve.

CD-ROMs. The sparse filling of the data matrix and the heterogeneous nature of chemical data pose potential problems for any retrieval system. To get to the desired data, often large intermediate hit sets have to be produced, which after intersecting are reduced to manageable proportions. With the advent of large, fast hard disks, cheap memory, and fast processors, these large hit sets, often tens but sometimes hundreds of thousands of records, pose fewer problems. Better, faster hardware has solved the problem. However, the use of CD-ROMs as a storage medium for chemical structure databases is not widespread,[1-3] and special considerations have to be taken to deal with these aspects. Most text retrieval systems for CD-ROMs assume relatively small hit lists and are thus not (necessarily) applicable for structure oriented chemical databases. The CD-ROM, a very practical, standardized, high capacity, cheap exchangeable medium has the disadvantage of having very slow disk access times. The increased spin rates have increased the data transfer rate proportionately, and it now nears that of a hard disk (Table I), so this poses less of a problem.

For many applications the CD-ROM is the ideal storage medium. They have a high storage capacity, 680 MB according to the present industry standards (but at the last count 720 MB is possible), low cost hardware, easy replication for production, low replication costs, and a well-defined standard format. However, with large databases, systems need to be designed carefully to ensure that the performance is acceptable.

Access Time. The access time is defined as the time taken to locate a piece of data on a CD-ROM. It is the sum of the seek time, the rotational delay, and the head settling time. Due to the constant linear velocity of the CD-ROM, it is also dependant on whether the data are to be found on an inner or outer spiral. Furthermore it is dependent on where the head was previously and on how easy it was to find the file in question.

Some of these factors can be influenced easily; the placement of files on the CD-ROM can be easily optimized, so that often used information should be placed as near the center as possible and files that are read consecutively are placed physically in the correct order. The directory structure should be a two level hierarchy with no more than ca. 40 files in each subdirectory. Avoidance of disk accesses can also be accomplished by using multiple storage techniques. Thus information that is required together should be stored together, even if this results in high redundancy, to minimize the number of disk accesses.

DATA STRUCTURES

To maximize the efficiency of a CD-ROM retrieval system, it must be ensured that the number of disk accesses and the amount of reading from the disk are minimized. The large capacity of a CD-ROM gives the possibility of the multiple storage of data necessary to be able to design a system that operates on a sequential basis and not a random one.

FACTUAL DATA

Hierarchical Data Structures. The logical data structure of the Beilstein database is hierarchical.[1] Thus facts are

* Abstract published in *Advance ACS Abstracts*, January 15, 1994.

0095-2338/94/1634-0032$04.50/0

Table I. Performance Comparison between a Hard Disk and a CD-ROM

	hard disk	CD-ROM
average seek time/ms	12	200
data transfer rate/(KB/s)	1200 KB/s	150/300[a]/600[b] KB/s
cost/dollars	1500/GB	2–40
exchangeable	no	yes

[a] Double speed. [b] Quadruple speed.

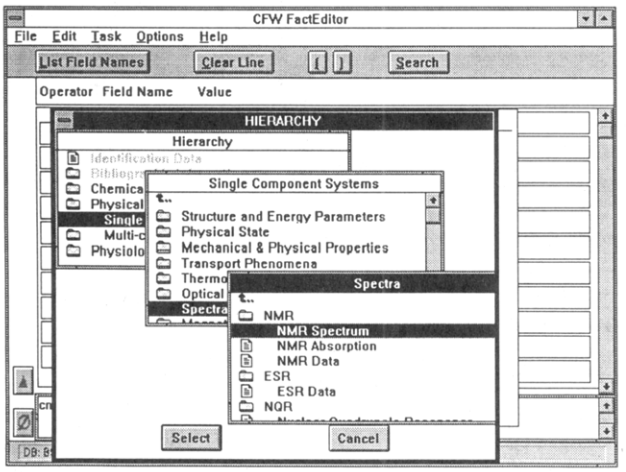

Figure 1. Hierarchical field code lists.

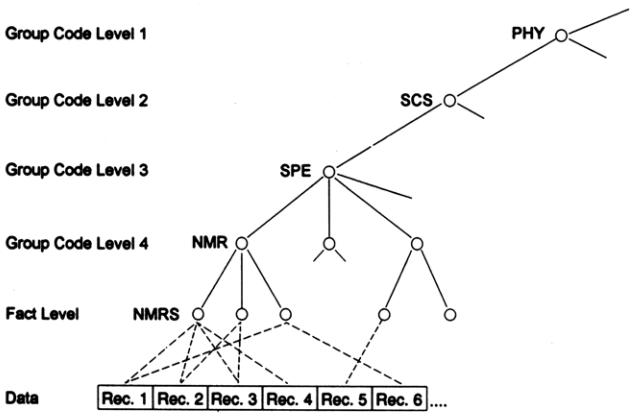

Figure 2. Relationship between the nodes in the data hierarchy and the sequential records of the database.

grouped together according to their type; for example, all spectral information is grouped under the SPE group term. This has been used in the on-line implementation as a way of ordering the search codes—the Beilstein database has over 350 facts—and a means of finding the search code in a list is vital. These terms are also used on display to allow the displaying of only certain types of information. In the Current Facts[1,2] implementation, this has been extended and these hierarchical terms are also searchable. The hierarchical field code list, used as a navigation aid, is shown in Figure 1. The relationship between the nodes in this hierarchy and the sequential records in the Current Facts database is represented in Figure 2.

Relational Data Structure. In some ways structure oriented chemical databases fit well in relational database structures. Each compound has various properties, and these can be stored in the tables of the database. There are however several problems: the main one is that much of the information is, or can be, potentially multiple. For example, the preparations of compounds can be carried out by using one or many starting materials. These starting materials are themselves database compounds, and to facilitate access, their registry numbers (primary key) also need to be stored with their chemical names in a field in the preparation table. Furthermore, each compound can have many data associated with it, and these data can be taken from different documents. Thus there must be pointers from the factual data records to the compound identification record and to the relevant citation record. Each citation can be pointed to by, or can point to, more than one compound. A relational database normalizes the data to store each piece of information only once. To allow multiplicity in subfields, one possible solution is to create enough fields in the tables to allow for all occurrences of the subfields, but this gives large unwieldy tables with many empty fields. Another solution is for further tables to be created, but this can lead to performance problems, particularly with the display, since to display one record (a compound) many tables have to be opened and read. A separate display file would allow fast display; however, highlighting could not easily be carried out, and some of the main advantages of relational databases such as updates and data consistency become more difficult.

Linear/Sequential Data Structures. A CD-ROM application cannot afford to carry out many random disk accesses and reads on different files for different tables necessary for one search or display. Thus a system has to be designed which does as much sequentially as possible. Thus the database must have very shallow indexes, and when searching, only the absolute minimum of information is read. An example of a suitable type of database structure is the balanced tree structure.

The Current Facts database is created in two steps, firstly a denormalization of the in-house database to give the data file. The data file has a sequential data structure; all the information for one compound is stored in one sequential record. Thus data which have been made nonredundant for the relational database, for example the citation information, must be stored explicitly—often multiply on the CD-ROM—in each record. This data file is used directly for display and is indexed in the second processing step to give the search file.

The Current Facts factual database requires the following of a retrieval system: the number of fields able to be indexed must be over 600; multiple fields and subfields must be allowed; numerical range searching must be possible with numerical values, which can be in the range of 10^{-99}–10^{99}, the usual text searching functions of wild cards, right and left truncation for words or phrases, comparison operators ("=", ">", "<"), and Boolean operators (AND, OR, NOT) are all required, as is the proximity operator—essentially equivalent to a sentence operator—to be used to tell if two subfields searched for are found in the same occurrence of a fact, for example, to determine whether the boiling point (BP) and boiling point pressure (BP.P) entries are present in the same occurrence of a fact. Without this operator many false drops would occur (since the AND operator only checks that the fields are present in the same record). Due to the large size of the database the storage requirements for the indexes must be less than that of the raw data. The response time must be short and rise less than linear with increasing hit list size. The system must be able to cope with large hit lists (100 000 records) and must be able to be integrated with S4-CD-ROM in all environments (DOS, Microsoft Windows, etc.).

Current Facts uses the Fulgor[4] retrieval system; this has a balanced tree structure based on inverted indexes to minimize the number of disk accesses. Reading has been reduced by compressing the index data very highly. The stored information includes the index terms and offset information for

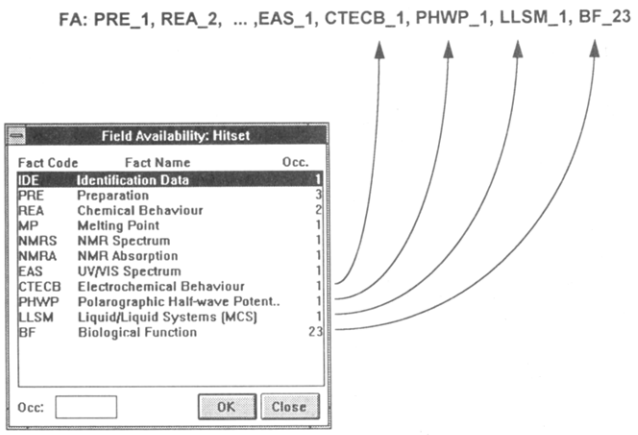

Figure 3. Schematic representation of the field availability field.

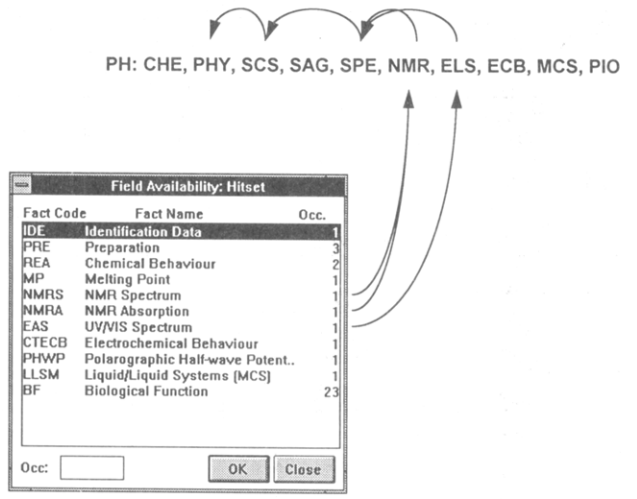

Figure 4. Schematic representation of the property hierarchy field.

the position of the record in the original file, the position of the indexed term in the record (necessary for highlighting), and the proximity operator information. When searching, only the absolute necessary information is read; the other information, related to display processing, is not read until required. Virtually all of the data are indexed; there are no stop words, and the ratio of raw data file size to index file size is about 2.3:1.

To display one record, only one sequential read is necessary. This is further optimized by compressing the data so that the amount of reading is minimized. This utilizes the performance advantage of the CPU over the data transfer rate of the CD-ROM. The compression ratio is about 2:1; the sum of the sizes of the compressed data file and the index file is approximately equal to the size of the raw data file.

Looped Data Structures. The Current Facts data structure is not purely sequential however; the internal cross references, brought about by, for example, the starting material field of preparations citing another compound that is also present on the CD-ROM, give a looped data structure. Access to these loops is provided by hyperlinks (see below).

MULTIPLE STORAGE FUNCTIONS

Field Availability. The ability to be able to search for the presence of data is a very useful function easily and efficiently implemented in a relational database. It simply requires accessing a particular table and reading the column of record numbers. It is often very useful to be able to search for all structures whose, for example, enthalpy of formation is known, without having to specify values. In systems which do not offer this function directly, the only way of achieving it would normally be a search from minus infinity to plus infinity (or the smallest value to the largest value), which can be very time consuming.

The solution adopted in Current Facts was to create a new data field, called FA (field availability). The contents of this field are the field codes of the fact fields (not subfields) of the record in question (Figure 3). This multiple storage technique could be used easily, and since the CD-ROM is never updated, just produced, there are no additional problems concerning updates and data integrity.

Thus searching for a field availability simply requires that a particular field code be searched for in the field FA. This is a very fast search. The field availability field is also used in the display to give an overview of the contents of a record. This can be used in the browsing mode, where there is not enough room on the screen to accommodate more than a summary, and in the full display, where this provides a flexible way of controlling the display that is particularly suited to records with many data.

Property Hierarchy. A similar method was adopted to implement the hierarchical search. The logical hierarchy of the data structure was previously only used as a means of grouping facts together in a printed list and as a means of selecting the display. As a search method this has uniquely been implemented in Current Facts. The PH (property hierarchy) field is created in an analogous way to the FA field. Thus the PH field contains the field code of the group terms for a record (Figure 4). To collect the required group codes, in effect a tree walk from the leaf to the root of the logical hierarchical tree (Figure 2) is carried out, writing all of the nodes into the PH field. This is actually done by using tables since the data are not stored in a hierarchical structure.

The hierarchical searching provides a very powerful method for general searches, a very important method of providing answer sets for browsing. For example, it is very easy to find all compounds with spectral data; the term SPE has to be searched for in the field PH.

CHEMICAL STRUCTURE DATA

The chemical structure data structure is inherently different from that of factual data. Each record is a structure, which is stored in one place, so that performance problems with display are not present. However, the traditional methods[5-12] of substructure retrieval can present large problems in a CD-ROM environment. When factual data are to be retrieved, it is sufficient to search the index to determine whether a hit is present or not. Chemical structure databases do not usually have an index in the same sense. There is often a screening step where structures are indexed according to fragments that they contain. However, the matching of one fragment from the query to that of one in the database does not mean that this structure is a hit. Other essential fragments could be missing, and the combination of the fragments in the stored structure may not match the query. Thus searching a structure data base is like a multiple term search in a factual database, whereby the original record has to be additionally checked to see if it really is a hit.

In the traditional two stage screening and atom by atom search (ABAS), this leads to an enormous amount of random disk accesses, with consequently unacceptable performance on a CD-ROM. The S4-CD-ROM[13] substructure search system has solved this problem in two ways: firstly, by including precise structural information in a hierarchical index,

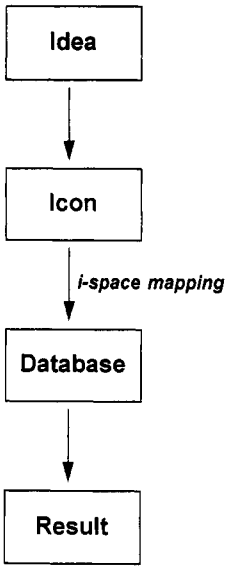

Figure 5. Searching in an ideal world using iconic mapping.

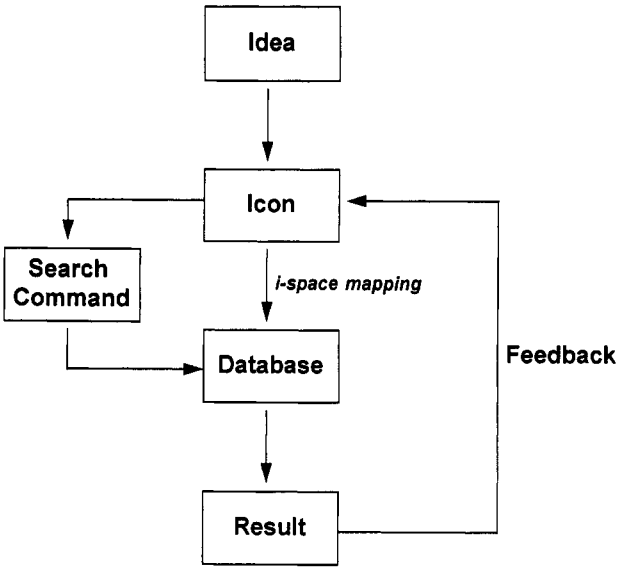

Figure 6. Searching in the read world, translating the icons into search strings.

thus increasing the screening efficiency and thereby minimizing the total number of atom by atom matches, and secondly, by multiply storing each connection table in the search file, effectively physically clustering the super-set of each answer set in the database. Thus from any bit-string screen only one seek and sequential read is necessary. The performance and the architecture of this system have been described elsewhere.[10-12]

INFORMATION ACCESS

In a chemical database, information access is the discovery of information about a particular chemical concept which can be defined in terms of a structure–property relationship.

Selection by Association. That we are constrained by the inflexibility of the present day databases and indexing systems has been realixed for a long time,[14] and various attempts at solutions have been made.[15] The way we think in terms of images, image mapping, and association is completely different to the way of thinking necessary for using a database. In an ideal world, we would have an idea, this would be fixed in a mental image or its iconic representation,[16] and this icon would be mapped onto the database that is created from real world icons (Figure 5). This i-space (iconic space) mapping would give us as exact an answer as possible and require very little modification. To find the best fit for an answer set would require some iterative processing, much like trying to find out the name of a species of bird we have seen by looking through a book containing pictures of birds. The present day ways of searching for information would no longer be necessary. Thus the ideal system would work with selection by association, which can be summed up by the following acronym WYTIWYG (*w*hat *y*ou *t*hink *i*s *w*hat *y*ou *g*et). The ideal system is still in the future, and we must contend with those which we presently have. This means translating the iconic representations into search strings, with inherent loss of information, searching the database and then almost certainly using feedback and modifying the search string to get a better answer (Figure 6). WYTIWYG does not yet apply.

Retrieval Efficiency. The efficiency of a retrieval system in allowing the user to get to the information required is something which needs to be optimized for each type of database. This is primarily a function of the indexing. Since in the Current Facts database each type of fact is indexed in an individual field, the problem has been largely moved to the interface, in that the user only needs to have an efficient way to get to the field code and does not need to be concerned about synonyms, etc., in the way that one does with a full text system.

Retrieval efficiency is generally defined in terms of precision and recall. Precision (P) defines the relevancy of an answer set and is given by the ratio of the number of relevant records retreived to the total number of records retrieved. Thus ideally this should have a value of 1.

$$P = \frac{\text{no. of relevant records retrieved}}{\text{total no. of records retrieved}}$$

Recall (R) defines the thoroughness of a search and is given by the ratio of the number of relevant records retrieved to the total number of relevant records. Ideally this would also have a value of 1.

$$R = \frac{\text{no. of relevant records retrieved}}{\text{total no. of relevant records}}$$

Retrieval efficiency (RE) is then defined as follows:

$$\text{RE} = P + R$$

In the factual database, by the virtue that all discrete facts are stored in separate fields, as opposed to a full text document based database, when searching for individual facts, precision and recall both have values of 1. This is also true for full structure searches in the structure database. Thus for these cases the retrieval efficiency has the maximum value.

The problems start when looking for compounds of interest in the database that could have a particular property. This usually involves carrying out a substructure search, where a good knowledge of chemistry and the retrieval system are required to ensure that all relevant answers are found. Precision and recall are also functions of the relevancy of the allowed methods of the specification of structure queries to chemistry.

The retrieval probability factor (RPF) defines the probability of finding a relevant record in a hit set. The definition assumes a randomly distributed database and answer set and is dependent on the precision and the maximum readable number of records (MRN).

$$RPF = (MRN - (1/P)) + 1$$

$RPF > 0$ the system and search are probably productive

$RPF \leq 0$ the system is overloaded

Thus, to diminish the change of overload, the precision has to be as high as possible and the number of readable records should also be large. The MRN is a function of the user interface. If an efficient browsing mode can be developed, then the user can comfortably look through large numbers of records. Since he will be often looking for similarities in structures, which are more easy to spot than similarities in full text documents, browsing is highly relevant to chemical databases and must be supported effectively by the user interface. The value of MRN is usually between 20 and 200, depending on the system.

TYPES OF ACCESS

For structure oriented chemical databases the types of access used can be classified as follows and can also be defined in terms of precision and recall.

Browsing. Unless the database is of minimal size that can be examined as a whole, or the data are sequentially sorted according to some useful classification system in the database, the term browsing can only sensibly be applied to the examination of general hit lists looking for a serendipitous hint for the desired concept. With average database sizes lying generally in the range of 10 000 to 10 000 000 records, browsing on the whole file is unlikely to be productive. The hit list can be arrived at either from the structure side or the factual data side (or a combination of both). Thus, for example, in a general substructure or similarity search, a hierarchical or wild card data search gives a hit list where the individual entries often have more information that is disparate than in common. Browsing through this list, just as one flicks through the pages of a reference book, relies on a combination of minimal presorting and serendipity to arrive at a lead.

Browsing in Current Facts has been made easier by firstly a very flexible display application, referred to as the short display mode. Structures and/or an overview of the factual information can be displayed in a variable size matrix. Thus six structures can be displayed at the same time. This not only makes browsing faster but also allows similarities between structures to be easily spotted. The second feature is the ability to select a subset of the hit list. Thus if from a particular list seven hits are of interest, selecting these with the mouse and then applying a filter will cause only these seven to be displayed. Perhaps more useful is the function that creates a subset hit list from the selected compounds. The means to do this by only using mouse operations is a significant advantage.

The retrieval efficiency for browsing is usually low; it is a probabilistic search method. However, this is not a large problem, since only a starting point for further investigation is required, not the complete information itself. The RPF tends often to be low, especially with large hit sets. Possible future work could involve the clustering of structures for selection of representative examples; this could ensure a better success rate.

Index Browsing. This is a very important and often neglected type of browsing. Simply examining the contents of an index can give not only information about the contents of the database useful to define the search space but can also be used to find information directly. Thus when carrying out combined searches, using Boolean operators, it is valuable to have some appreciation for the overall contents of the numerical fields before searching. In text fields, index browsing can be used to access information. In the Current Facts database, the contents of the Biological Function field provide one of the best examples. Examination of the indexed items can give suggestions for possible interesting compounds. This has been implemented as an interrupted hyperlink (see below).

Navigation. This is defined as targeted examination of a database by traversing from one node to another. It is an explicit recall method. In a chemical database the best example would be that of a synthetic pathway;[17] moving backward or forward in the pathway is achieved by jumping to the referenced starting materials or product of a chemical reaction by using a hyperlink. Thus the user clicks on a highlighted hyperlink, and the user interface automatically takes care of any processing, allowing the record pertaining to the link to be immediately displayed. Navigation is a direct link; thus recall and precision are both 1.

Linked browsing. This is the browsing of a database using hyperlinks. This is a combination between navigation and browsing where the path is not defined as in navigation. The links between records are explicit rather than probabilistic. The links between starting point and target is, however, probabilistic. Recall and precision can be between 0 and 1.

Feedback Searching. Feedback searching involves modification of the query to get to the closest matching answer set. It is mainly required due to the lack of the i-space searching capabilities. Feedback searching provides a query driven means of improving the results and often involves the balancing of precision and recall. However, the final answer set does not have to be a subset of the original query answer set. Thus the RE can be anywhere between 0 and 2; the crucial point is that RPF must be high enough to be able to find examples of relevant hits in the answer set.

HYPERLINKS

Hyperlinks[15] are present in most present-day chemical database systems. An analysis of their function shows that the set, as defined below, is often incomplete; thus some functionality is missing. The Current Facts data structure, in giving a field or subfield to each fact, makes it easy to create hyperlinks. Instead of having to analyze the contents of a text record for possible links and then code them, perhaps by hand, the hyperlinks can be coded by field presence, not content. For example, the content of the starting material Beilstein registry number (BRN) field can, by definition, only be a BRN; not only that, but by definition, this BRN must also be present in the database. Thus to set up the link, the only requirement is the presence, in a record, of the PRE.SB subfield.

Hypertext Links. This is defined as a link from text to text. An example would be the name of a chemical compound providing the link to the full record for that compound.

Hypermedia Links. This is defined as a link from one sort of media to another. The usual example would be from text to an image and vice versa. In the chemical database the example would be from text (chemical name or registry number) to a graphical structure. The usual link would be a name or registry number to the structure.

The above two types of links can be further subdivided into classes, depending on the relationship that the link provides between the records:

Standard Hyperlink. This is a one to one link. From the point of view of the purist this is the only true hyperlink, although the other terms can be justified to be brought into

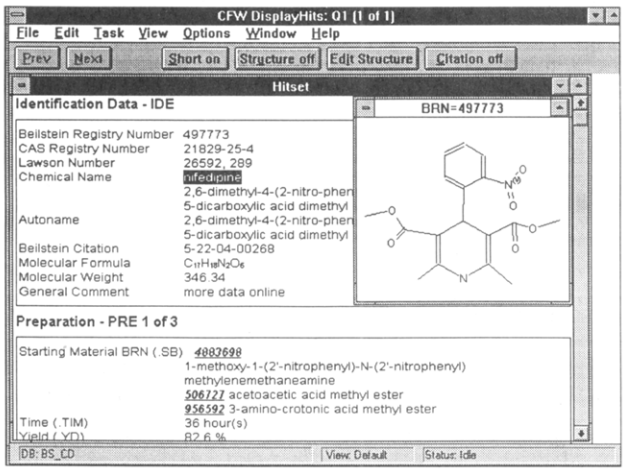

Figure 7. Highlighted starting material hyperlinks in the display of nifedipine.

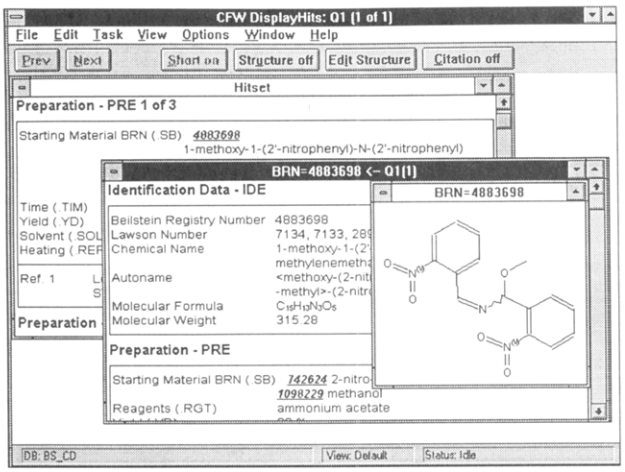

Figure 8. First level in the synthetic pathway to nifedipine, displayed by activating a hyperlink.

the classification due to their similar functionality. In Current Facts the most important link falls into this class. This is the link between a referenced compound, as cited in the starting material BRN field for example, and its record. The link is provided by the presence of the PRE.SB subfield. If it is present, it will be highlighted in the display (Figure 7), and when the mouse is over the field, the mouse pointer is transformed into a hand to signify that this is a hyperlink. Clicking the link immediately displays the full record of the referenced compound (Figure 8). This can be repeated as often as required, by clicking each time the PRE.SB subfield in a newly displayed reference compound, until as much of the synthetic pathway as required is displayed (Figure 9). This provides a high degree of functionality to synthetic path navigation and is not document bound.

Branched Hyperlink. This is a 1 to *n* link, where *n* is between 1 and the number of records in the database, but to be practical should be under ca. 50. The most useful example is that of citations. In the Current Facts structure oriented database, the original citations, containing on average seven compounds, are split into seven discrete records. Each record contains information from potentially more than one document. Thus to preserve the compound–document information, each fact has to have a pointer to its document source. Thus to recreate the document, the citation has to be searched for. This function could be provided by a hyperlink. Thus clicking a citation would result in the generation of a display containing all of the compounds cited. At the moment this is achieved by

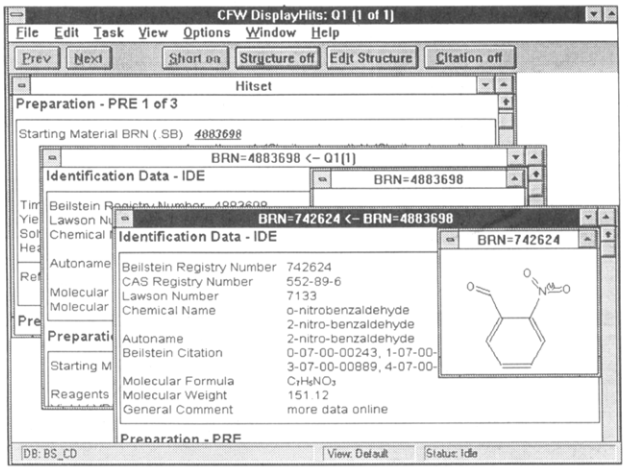

Figure 9. Second level in the synthetic pathway to nifedipine, displayed by activating a hyperlink.

copying the citation into the fact search mask, via the clipboard, where it can be searched. Since no typing is required, an accurate transcription is guaranteed.

Navigation through the nodes of the hierarchical data structure can be carried out by using the hierarchical field code list (Figure 1) option in the Current Facts user interface. Clicking one of these codes transfers it to the fact editor where a field availability of property hierarchy search can be started. This is effectively a branched link.

Interrupted Hyperlinks. These are not strictly hyperlinks; but their function is highly similar to that of hyperlinks, involving the displaying of additional, related information that is accessed from the displayed record.

An example of an interrupted link is the copying of a structure from the display to the structure editor for modification before searching. One requirement for a hyperlink is the necessity for mouse-only operations; no rekeying of the information should take place. A structure can be transformed into a substructure with only a few mouse operations. This could of course be carried out automatically and then be implemented as a true link. However the user control, especially with chemical structures—since what is required is usually a substructure which is present in the displayed structure, and not that the displayed structure be a substructure itself—can be essential to the retrieval efficiency.

Further examples can be made with data. Index browsing can be carried out with the Current Facts user inferface, and the records associated with a term of interest can be retrieved by simply clicking the displayed term with the mouse. The index term is copied into the fact editor, where it can be edited, for example, truncated and given wild cards, and then searched for.

Intelligent Hyperlinks. These types of links can be envisaged as occurring at various levels of intelligence, simply, as with the automatic optimization of the above citation query, which requires more the functionality of an expert system, and similarly, with the use of a thesaurus to be able to link to similar objects. Higher level intelligent links have the aim of achieving selection by association, retrieving the answer you thought about and not necessarily that which you would have retrieved when searching using the indexes. The definition also includes user-defined links; the user can define a link between objects, for whatever reason, and that link is usable at a further date, the intelligence being supplied by the user and not the system.

CONCLUSION

This paper has described the way a system has been optimized to search on a CD-ROM to maximize the performance. The functionality of the user interface and its important role in the access of the inherent information contained in a database were described. Further work on intelligent functionalization of systems would allow selection by association be brought closer and hence increase information access.

ACKNOWLEDGMENT

The author would like to thank Helmut Grotz for helpful discussions and to acknowledge the work of the software developers and chemists who worked on the Current Facts project.

REFERENCES AND NOTES

(1) Hicks, M. G. Beilstein Current Facts in Chemistry: a Large Chemical Database on CD-ROM. *Anal. Chim. Acta* **1992**, *265*, 291–300.
(2) Hicks, M. G. Similarity and the Beilstein Information System: Searching for Concepts with Current Facts. *J. Chem. Inf. Comput. Sci.* **1992**, *32*, 631–638.
(3) Warr, W. A. New Chemical Databases on CD-ROM. *Database* **1993** (Feb), 59–67.
(4) Fulgor Retrieval Software. Running Bytes GmbH: Niedstrasse 22, 1000 Berlin 41, FRG.
(5) Willett, P. A. Review of Chemical Structure Retrieval Systems. *J. Chemomet.* **1987**, *1*, 139–155.
(6) Willett, P. *Similarity and Clustering in Chemical Information Systems*; Research Studies Press Ltd.: Letchworth, Hertfordshire, England, 1987; pp 10–18.
(7) Ash, J. E.; Chubb, P. A.; Ward, S. E.; Welford, S. M.; Willett, P. Chemical Structure Search Systems and Services. *Communication, Storage and Retrieval of Chemical Information*; Ellis Horwood: Chichester, U.K., 1985.
(8) Stobaugh, R. E., Chemical Structure Searching. *J. Chem. Inf. Comput. Sci.* **1985**, *25*, 271–275.
(9) Barnard, J. M. Problems of Substructure Searching and Their Solution. In *Chemical Structures*; Warr, E. A., Ed.; Springer-Verlag: Berlin, 1988; pp 113–126.
(10) Hicks, M. G.; Jochum, C. J.; Maier, H. Substructure Search Systems for Large Chemical Databases. *Anal. Chim. Acta* **1990**, *235*, 87–92.
(11) Hicks, M. G.; Jochum, C. Substructure Search Systems. 1. Performance Comparison of the MACCS, DARC, HTSS, CAS Registry MVSSS, and S4 Substructure Search Systems. *J. Chem. Inf. Comput. Sci.* **1990**, *30*, 191–199.
(12) Bartmann, A.; Maier, H.; Walkoviak, D.; Roth, B.; Hicks, M. G. Substructure Searching on Very Large Files by Using Multiple Storage Techniques. *J. Chem. Inf. Comput. Sci.* **1993**, *33*, 539–541.
(13) S4-CD-ROM. Softron GmbH: Rudolf-Diesel-Strasse 1, 8032 Graefelfing, FRG.
(14) Bush, V. As We May Think. *Atl. Mon.* **1945**, *176* (1), 101–108.
(15) Woodhead, N. *Hypertext and Hypermedia; Theory and Applications*; Sigma Press: Wilmslow, England, 1990, and references cited therein.
(16) Aleksander, I. Personal communication.
(17) Hicks, M. G. Reactions in the Beilstein Information System: Nonaporic Organic Synthesis. *J. Chem. Inf. Comput. Sci.* **1990**, *30*, 352–359.