A United Kingdom Approach to Legal Information: the STATUS Project

N. H. Nunn-Price


1. SOME BASIC CONCEPTS

All organised societies operate within the framework of a legal system. Any such system will reflect the characteristics of the particular society from which it grew, so it is essentially national in character. Any computer system designed to work in a legal environment will have to take into account the national flavour, the nuances of the legal system.

In a review of legal computer systems developed in Europe, two main types of system can be seen to have emerged. In the countries of southern Europe, where the law is codified, and where there is doctrine, the emphasis has been on systems that handle structured information stored in classified fields, often using a specified vocabulary of keywords. Much of the analysis required is done intellectually before entry into the computer, and this approach has been supported by the design of thesaurus-type aids to searching.

Norman Price has been involved in legal information retrieval since 1968 when, with Professor Bryan Niblett, he started the STATUS Project. He has been a member of the Council of Europe Committee on Legal Data Processing and is currently attached to the European Commission in Brussels where he is the Project Leader for the Technical Study in Legal Information Retrieval.

* This paper discusses some of the reasons which have led to the development, in the United Kingdom, of full text retrieval systems in the field of legal information rather than systems operating on previously analysed information. A description of the STATUS system is given and attention is drawn to the need to design systems that can work with a mixture of integral text and structured material.
In contrast, in northern Europe (particularly the United Kingdom), the law is heavily based on Case Law, or Common Law, in which the meaning of a statute is tested by argument in court, and where there is little or no codification. This has led to the development of computer systems that operate on the full text of Statutes or Cases. The emphasis has been on facilities for searching, using words and phrases in natural language, to retrieve the original text of the legal documents required. This is radically different in approach from that of structured information, and has been slower to develop because of the complexity of the programs and the cost of preparing the full text in machine-readable form; and the further cost of the disk storage needed. These costs have diminished considerably in recent years, so this type of system has become more economic, and a number of systems are in operation. This paper will, in passing, describe one such system – the STATUS system – in commenting on the characteristics of these systems.

What is now of great interest is that the advent of international legislation, especially by the E.E.C. in Brussels, has imposed a need for the two approaches – that of structured information, and that of full text – to meet on common ground. Out of this fusion the next generation of retrieval systems will arise, for it is clearly necessary to develop techniques that will deal fluently with information which is a mixture of free text and structured fields.

2. The United Kingdom Legal Environment. Origins of the STATUS Approach

It is useful to consider the United Kingdom legal environment as a document-centred system, since some form of printed text is the start, the continuation, and the final outcome of any legal process. It is much more than that, of course, for it must never be forgotten that people are deeply involved; but for our purpose it is convenient to consider the British legal system as one based on words contained in documents. There are many different opinions on the fairness, effectiveness or efficiency of the U.K. legal system, but there is one point on which there is total agreement: that there is an enormous quantity of legal data now existing, which is growing rapidly in both size and complexity so that it threatens to engulf professional practitioner and layman alike in a remorseless tide of paper. Traditional methods used for the organisation and dissemination of legal information have reached their limit, and new techniques are urgently needed. Fortunately, the stored program computer has come along at the right time to provide a solution.

It was in this context that STATUS began, in order to apply computer techniques to the storage, analysis and retrieval of legal information,
with particular emphasis on Statutes enacted at Westminster. The acronym STATUS is derived from STATUtE Search. Work on the project began in late 1967 with a feasibility study which gave a favourable opinion on the potential application of computer techniques in this field. Initially, the project was simply a research topic, but as time progressed working solutions became practical reality. From the beginning, opinion was sought from legal practitioners, and the current system owes much to the feedback of ideas and comment from the end users. This is essential in any system that is intended to operate in real life. Early in the life of STATUS, four design decisions were made which determined the course of the project and outlined the philosophy to be followed. These four decisions were:

i) That the computer programs should be written in a high-level language in order that the system should be machine independent as far as possible, and certainly so that the logic and flow of the programs could be well understood. The communication of ideas was part of the approach adopted. Several languages were tried, but the system inevitably settled for ANSI Fortran IV - not because it is the most suitable, but because most computer installations can accept Fortran, and portability of the system was regarded as a major objective. There are currently working versions of the STATUS system on IBM, ICL, Honeywell, DEC, Rank Xerox and CII machines.

ii) That the full text of legal documents should be processed so that the lawyer would be sure of access to the original source document, and not to a précis or an abstract. Information changes in value with time, and no one summarising a legal document can possibly anticipate how a particular piece of legislation will be used and interpreted in future.

iii) That the essence of the problem was how to get a good answer from a bad question. The solution was to work on the question, successively refining it until the right question has been formed, and the correct information retrieved. The best way of doing this is to carry out a dialogue with the database, to interrogate it by searching interactively.

iv) That the system should be simple to use, so that any reasonably competent person, unskilled in computing, would be able to operate it. Commands, instructions and questions should be in words and phrases in natural language. The computer terminal should be as easy to use as an electric typewriter: in short, the system should suit the user.

These four criteria have stood the test of time, and still exert considerable influence on the current version of STATUS. One additional factor has been added to the above list, and that is the ability to amend and update the databases dynamically: that is, interactively from the terminal while the system is operating normally. This ensures that the data stored is kept fresh and up to date.
3. Preparation of the Source Text

The starting point of any full-text system is the preparation of the text in computer readable form. This apparently mundane task is expensive and time-consuming, but is of vital importance since the ultimate system will be judged on the quality of the text retrieved. Accuracy is at a premium. STATUS has accepted data prepared by a wide variety of methods, including punched cards, key to disk or magnetic tape, optical character recognition, and – more recently – magnetic tapes generated during computer typesetting of the source documents. It is the advent of this last technique that has made the full-text approach more economic, since these tapes are a by-product of the publishing process with the further advantage that the quality of the information recorded is very high because of the traditional craft and skill of the printing industry. The printers’ symbols stored on tape are also useful to the programs analysing the text. Ideally, any text for entry into a computer system should be captured at source and keyboarded once only, but any practical system will have to put a substantial backlog of material into machine readable form. This is a «once-for-all» capital cost, and should be charged as such. Optical character recognition – a type and scan operation – has proved a satisfactory method of putting large amounts of old information into STATUS.

When the text is ready it may be fed straight into the system. One of the main advantages of the full-text approach is that there is no need for the expensive, time-consuming and intellectually demanding effort required for the indexing and classification of the documents to be stored. The raw text is fed in and the computer programs do the rest.

One final point is that legal text changes frequently; the law is very much alive, so there is often a need to alter the text. An on-line editing facility is an essential part of a working system which is why STATUS now has such an editor as an integral part of the package.

4. The Components of the System

Any information retrieval system has three main components, namely:

1) the computer programs;
2) the databases;
3) the users;

each of which is distinct but dependent on the others, and each of which is of equal importance in the design of a working system. Here we are mainly concerned with the programs, which fall conveniently into two groups: firstly, those which construct the data base; and, secondly, those which provide the searching capability.
It should be noted that decisions made during the generation of the data base from the full text will determine the strategies that are available in the search process. Once information has been deleted, it cannot be recovered.

4.1. Construction of the data base

Text is composed of words written to convey meaning, with no thought of the problem of retrieval; the words occur in a random order so that the basic task of this group of programs is to put the words into an ordered sequence. This is normally alphabetical, as in STATUS, but need not be so. The end product of this analysis is known as a 'concordance' or inverted file. Each different word in the text is stored, together with the references to each and every occurrence of it. These references give the precise address of each location of a particular word in the text. In STATUS each reference has four attributes: chapter, article, sentence and word, which together define a unique location. The attribute chapter specifies a large segment of the text, such as an Act of Parliament or a complete Treaty; an article is a logical unit of information within the chapter – for example, a Section of an Act, a case report, a page, or whatever the user decides is best in a particular application. The article is usually the retrieved answer to a posed question, and is usually displayed at the terminal. The sentence attribute is simply the ordinal number of the sentence within the article; and finally the word attribute gives the position of the word within the sentence. If required, a paragraph may be specified instead of a sentence. The user can vary these definitions to suit the type of text being analysed. Concordances are quite useful, and in themselves form a handy reference which may be printed and published. A typical entry is shown in fig. 1.

![FIG. 1](image)

The size of an inverted file, with all the different words and the references, will be roughly the same size as the original text. Since the text itself will also need to be stored, then the total space required is about that for the text alone, and the size of the concordance should therefore be reduced if possible. Not all words contain useful information: some, such
as "the", "of", "to", "for" and so on, are simply links in the construction of the text, so there is no point in storing their references. Since these tend also to be the most frequently occurring words, a substantial saving of storage space can be made by suppressing the references to such words (while keeping them in the original text). In English, about 30% of storage can be saved by suppressing references to 24 of the most common words; the same is true of other languages. A judicious choice of these so-called common words can reduce the space required by the inverted file to about half of that required to store the original text.

The program that creates the inverted file may be called a concordance generator, and is fundamental to any full text system. It must be efficient and comprehensive. In STATUS this program has been made portable by adopting a modular construction in which those minor parts of the program (such as character handling), which tend to be machine dependent, are isolated in separate subroutines which have a defined interface to the main program. These short subroutines may easily be modified when required.

Once a concordance has been generated, it may be stored permanently on disk or tape. For an interactive searching system, a direct access file copy of the concordance will be stored on disk, and an appropriate index created that will point to the disk addresses where the component parts of the concordance are stored. Likewise, a random access file of the source text will be created, and the two files will be related through sets of pointers. In this way a complete data base is constructed and made ready for searching.

In considering the data base, mention should be made of the amending and updating problem. In the early versions of STATUS, this was done as a separate operation in batch mode, but in the current version (known as STATUS II) this can be carried out interactively from a terminal. This has necessitated a more complex structure of the concordance, which will be discussed later.

4.2. The searching component

Since no information is lost in the storage process, apart from the references to the common words, then all the attributes stored can be used in the search process. The development of search languages and their associated techniques is one of the most fascinating problems in information science, and one that is likely to engage us for many years. In STATUS a search language has been developed that is flexible enough and powerful enough to scan through the data base and find the required information. A good deal of effort has been put into making it appear natural to the user.
Words in text have two basic relationships which are: 
a) their position with respect to each other, especially with other words immediately adjacent; and 
b) their logical relationship which may be specified by the operators « and », « or » or « not ». The simplest search is for a single word - e.g.

income?

which, if applied to financial legislation, would yield a vast number of references. This question could be modified by the use of the logical operator « and » thus:

income and tax?

which conveys a more precise meaning, and would reduce the number of answers. Such a question can be built up to any degree of complexity. For instance,

income or salary and tax not dividend?

which shows the use of the three logical operators. These operators act across a logical unit of information (article), so the terms can be quite distant from each other in a long article. The search may be further tightened by using the relationship of position which normally is limited to a sentence or paragraph. Thus:

income and (+1) tax?

specifies that the term « tax » must be in the same sentence as « income » and immediately after it. The information content of pairs of words, or longer groups of words, is so high that it is worthy of additional programming to allow natural combinations of words in phrases to be used in questions. For example,

income tax?
value added tax and current rate?

In these phrases the logical and positional operators are implied, and the user need not be aware of using them. The program can insert them internally.

Common words may be of little value for storage in the inverted file, but they do help the average user in framing a question; therefore the search language should allow the use of common words, and deal with them internally. A question such as:

the right to withdraw from the treaty?

which conveys an important concept and uses several common words, should be perfectly valid.

Another useful function is the ability to use stems of words when searching. This truncation facility is very useful in dealing with plurals,
and is particularly valuable for languages which are highly inflected, such as French or Italian. For instance,

author®?

would find all references to

author, authors, authorisation, authority, authorities.

All text has form and structure, particularly if it is specialised text, so a facility which enables the user to take advantage of his or her own specialist knowledge of the subject matter of a data base can be a powerful aid to searching. The « macro » function in STATUS enables frequently used standard searches, or very complex searches, to be stored in a user library, and be called by name when required. These macros may contain logical functions of an required complexity.

A sample of STATUS search dialogue, including the use of a macro, is shown in Appendix 1.

The automatic response from the system to a question is to give the number of documents in the data base that satisfy the question. The enquirer will then decide whether to display the titles of these documents and/or the original text of the documents. Any individual document may be displayed on the terminal. At this stage the enquirer is dealing with the text file, so a natural thing to do is to browse through the text, beginning with an article that satisfies the original search, and then browsing backwards or forwards through the text following cross-references or a train of thought started by the answer. It should be noted here that the enquirer has left the bounds of the original search, and is doing the computer equivalent of turning the pages of a book. This facility is useful in browsing through Statutes which tend to include in their drafting a complex network of cross-references.

The search process so far described works on the semantic ideas and concepts expressed in terms of the logical and positional combination of words and phrases. It pays little attention to syntax, and crosses the boundaries imposed by paragraphs and sentences. This method has proved remarkably powerful, and has been more successful than originally expected.

5. **Outline Description of STATUS II**

STATUS I was designed with fast on-line searching of large text data bases as the prime requirement – and the creation and updating of the data bases was treated as a separate, distinct function, carried out in batch mode. There are, however, applications where both fast searching and
rapid amendment of the data base are needed. STATUS II is a response to that requirement, but also embodies other facilities desired by users, including:

a) built in security keys in the software;
b) a single command language throughout the system;
c) the ability to carry out all operations from the same terminal;
d) the ability to search a variety of information in different formats and structures.

The new system has coalesced the searching and storage components of the programs into a continuous spectrum of operations.

The system will run in any of three modes:

CREATE
ENLARGE
SELECT.

The CREATE mode is intended for the creation of new data bases. The ENLARGE mode is somewhat similar to CREATE, but is specifically intended to control the addition of new data, which may be of large bulk, to an existing data base. The SELECT mode makes available all facilities for search and retrieval, and also provides the ability to modify the existing contents of data bases.

5.1. The Command Language

The Command Language operates across three modes. There are six groups of commands, namely:

1) Initial commands
SELECT, CREATE, ENLARGE

2) Global commands
END, BREAK, NOVICE, EXPERT, HELP

3) Common Word commands
LIST, INCLUDE, EXCLUDE, ACCEPT, REJECT

4) Search commands
QUESTION, CHAPTER, CONTENTS, SET, MACROS, THE QUEST SEARCH LANGUAGE

5) Examine commands
DISPLAY, TITLES, FORWARD, BACK

6) Revise commands
DESTROY, EXTEND, FOLLOW, ACCEPT, REJECT, UPDATE, ADD, EDIT (DELETE, INSERT, REPLACE, LIST, REVERT...), REGISTER

All these are available to the user at the terminal, but there may be some restrictions because of the status accorded to a particular user. These constraints apply to a particular data base, and the idea of a « manager » is introduced to control the power to amend a data base. Note the inclusion of an EDIT command. An on-line editing facility has always been regarded as essential in text-processing, so the logical step was to incorporate an editor in the STATUS system.
5.2. Text structure of the input text

Although STATUS II is not limited to natural language text, the inherent structure of such material forms a useful basis for organisation. The data base is defined at four levels:

Chapter
Article
Sentence
Word.

Of these, the fundamental level is that of «article», which may be defined as a logical unit of information. Articles may be of any length, and the division of the text into articles is specified at the time of preparation of text for input. The response to a search is a set of relevant articles. The division into articles should reflect the inherent structure and thematic content of the text, and should also take into account convenience of length for retrieval. It might be based on paragraphs, groups of paragraphs, case notes, records, tables, sections, etc.

A «chapter» is a sequence of articles, generally related as in a Treaty, Act or Case, and chapter boundaries are fixed when the data base is created. Searches may be limited if required to within particular chapters rather than a whole data base.

In natural language text, the terms «sentence» and «word» have the normal meaning. In the more general case (e.g. structured information), a «sentence» is a delineated subdivision within an article, and a word is simply any string of characters beginning and ending with a blank. In addition there are embedded security keys which can descend to the level of an individual word.

The manner in which information is stored enables searching to be within data bases, within chapters, or across other specified categories.

5.3. The structure of the data base

In any single data base there are two files, first the source text with embedded titles, and second the concordance or inverted file of that text. Both of these files consist of chained blocks of data (see figs. 2 and 3).

The text file will contain, at any instant, several types of text, including:

i) the text of articles which have been concorded; this text will include a flagged title to the article;
ii) the text of articles which are being amended;
iii) the text of articles on which amendment has been completed but the new version of which has not yet been entered in the concordance. These articles are checked for conformity with the text directives and then entered into an «amendment list». The command UPDATE sends these
FIG. 2. – STRUCTURE OF THE DATA BASE
articles to be processed internally by the concordance generator, and these become:
iv) concorded articles chained at the appropriate place in the text file, and finally removed from the amendment list.

While this is going on, signals are set to warn anyone searching that article that it is about to be amended.

The concordance file is likewise composed of chained blocks. Entry to the file is via direct access into a matrix composed of the first pair of characters of any text word or term (fig. 3). From the matrix there are pointers to fixed blocks containing all the possible second pairs of characters in the words. The entries in these blocks point to variable length blocks containing the remaining characters of particular words, which then point to the concordance reference blocks containing the sets of four-level references, which give the precise location of each occurrence of the word.

The entry matrix is held in core, but all the other blocks are on disk so that the pointers are block pointers to blocks held on disk within a defined file space. A new term is entered by skipping down the tree, matching characters where possible, and then inserting new ones where required, and finally putting in the references.

5.4. Amending the data base

The two kinds of updating which may be distinguished by volume are a) editing updates for small changes, and b) bulk updates for large quantities of text. Editing updates are small amendments and deletions carried out by authorised users. These are first entered into the text file as already described, and then the concordance modification is initiated by the UPDATE command. This entails breaking and remaking the chains where necessary, altering the pointers, checking security keys, and finally putting in the references. In processing such amendments, a number of deletions may be made, and empty records will be left in some of the blocks. These empty records will remain until such times as a «house-keeping» program is run to compress and tidy up the concordance.

Bulk updates are normally carried out by registered «managers». This is the type of operation that creates new data bases from source material, such as magnetic tape, or that carries out large scale amendments of an existing data base. In essence this is a remote job entry method of working where the terminal is used to initiate and control the required program.

It may be noted, in passing, that there is no difference in treatment between a search question or an amendment so far as access to the concordance is concerned.
FIG. 3. - STRUCTURE OF THE CONCORDANCE DATA BASE
5.5. Resource usage

The design has been based on a data base of 16 million words of English, with an estimated vocabulary of 160,000 different words. Assuming an average length of six characters per word, plus one extra for the blank at the end of the word, then this text would occupy 112M bytes of disk storage. The addition of block headers, chapter pointers and indexes would add an estimated 5M bytes to this figure. A block size of 256 bytes and an average article of, say, 500 words, would require 15 disk accesses to be brought out entirely, but this operation would be dominated by the speed of the terminal.

The concordance would have an average of 400 references for each different word. These references could be packed in a variety of different ways. For example, assuming a maximum of:

- 1024 chapters - 10 bits
- 1024 articles per chapter - 10 bits
- 128 sentences per article - 7 bits
- 128 words per sentence - 7 bits

then one reference could be held in 34 bits between 4 and 5 standard 8 bit bytes. If there are no common words, and all references are left, then this concordance would require 80M bytes. If there are designated common words, this figure would reduce by 40% to 48M bytes. The total size predicted for 16 million words is between 180→190M bytes, or 150→160M bytes if common word references are removed.

6. Conclusion

Any practical information retrieval system intended to function in a normal working environment will contain design compromises in order to meet conflicting requirements. An inverted file is designed for rapid searching, but is difficult to amend, while a sequential text file is easy to edit but cumbersome to search. The structure chosen is an attempt to reconcile these two extreme conditions. In STATUS I the response time is extremely rapid, even for complex searches, and is determined mainly by the speed of the terminal. Internally, there are long periods when the CPU is inactive and waiting. In an average of number of one-hour sessions with STATUS, it is found that only 15 seconds' CPU time is used, most of the time being spent in moving data to and from disk. In STATUS II this time has been put to use in the amending process.

In the wider field, it is now apparent that full text retrieval systems have become a practical alternative to structured systems, and furthermore have
considerable advantages. The future trend lies with a combination of structure and free text, which may be called amplified full text, and this demand will bring the two different approaches together.

BIBLIOGRAPHY


APPENDIX 1

Sample of STATUS II dialogue

q tax?
QUEmON IS SATmFIED BY 135 ARTICLES

q tax + purchaser of property?
***WARNING-OF IS A COMMON WORD
QUEmON IS SATmFIED BY 1 ARTICLE

t

*Finance Act 1975 S. 26, Exceptions from liability.
(1) A purchaser of property, and a person deriving title from or under such a purchaser, shall not by virtue of subsection (2)(b) or (5)(c) of section 25 of this Act be liable for tax attributable to the value of the property, unless the property is subject to an Inland Revenue charge.
(2) No person other than those liable under subsection (1) or (2) of section 32 or subsection (3) or (4) of section 34 of this Act or, in MORE? REPLY YES, NO OR QUIT :

no

q value added tax?
QUEmON IS SATmFIED BY 6 ARTICLES

t all


4. *Finance Act 1975 S. 4, Refund to diplomatic missions, etc., of MORE? REPLY YES, NO OR QUIT :
yes V.A.T. on importation of hydrocarbon oil.

5. *Finance Act 1975 S. 59, Citation, interpretation, construction and repeals.


q close company, companies?
QUEmON IS SATmFIED BY 13 ARTICLES

q close company + transfer of value?
***WARNING-OF IS A COMMON WORD
QUEmON IS SATmFIED BY 2 ARTICLES

t all

2. Close companies *Finance Act 1975 Sch. 5, P. 24. (1) In relation to a person who is a participant in his capacity
APPENDIX 2

Example of a macro showing its use and also a list of its components

> c .define, barrister ?
   QUESTION IS SATISFIED BY 1 ARTICLE
>
d

Supplemental *Finance Act 1975 S. 51, Interpretation of Part III.
(1) In this Part of this Act, except where the context otherwise requires, - "amount" includes value; "barrister" includes a member of the Faculty of Advocates; "the Board" means the Commissioners of Inland Revenue; "charity" and "charitable" have the same meanings as in the Income Tax Acts; "disposition" includes a disposition effected by associated operations; "enactment" includes an enactment of the Parliament of Northern Ireland and an Order in Council made MORE ? REPLY YES, NO OR QUIT :-

no
>
body define
(NEAN*, DEF IN*, DEEM*, CONSTRU*, INCLUD*, PURPOSE*)/-12,12/
>
end

121
APPENDIX 3

Example of the use of the dynamic editor

> q insurance
> QUESTION IS SATISFIED BY 4 ARTICLES
> t all
> * 1. RETIREMENT PENSIONS (TAXATION)
> 2. TRADE, INSURANCE COMPANIES
> 3. TAXATION (SOCIAL INSURANCE BENEFITS)
> 4. MINISTERIAL RESPONSIBILITIES
> d 1

RETIREMENT PENSIONS (TAXATION)

Mr. William Hamilton asked the Chancellor of the Exchequer if he will consider the relief from taxation of all retirement pensions payable within the National Insurance Scheme.

Dr. Gilbert: I have noted my hon. Friend's suggestion but I do not think it would be justifiable to relieve from tax all national insurance retirement pensions, regardless of the size of the pensioner's income.

> edit
E>
rep/National Insurance Scheme/National Pension System/
E>
list
££T
RETIREMENT PENSIONS (TAXATION)<
££T

Mr. William Hamilton asked the Chancellor of the Exchequer if he will consider the relief from taxation of all retirement pensions payable within the National Pension System.<

Dr. Gilbert: I have noted my hon. Friend's suggestion but I do not think it would be justifiable to relieve from tax all national insurance retirement pensions, regardless of the size of the pensioner's income.<

££A
E>
check
CHECKED
EDIT, ACCEPT OR REJECT
```

122