The DataLex Legal Workstation - integrating tools for lawyers

Graham Greenleaf, Senior Lecturer in Law, University of New South Wales
Andrew Mowbray, Senior Lecturer in Law, University of Technology, Sydney
Alan Tyree, Landerer Professor of Information Technology and Law, University of Sydney

1. Integrated computerisation of law

Computerisation of law has developed from a number of originally unrelated technologies: the development of online free text retrieval systems from the 1960s; the revival of artificial intelligence research in the form of expert systems in the 1970s, the related development of automated document generators, and the 'rediscovery' of hypertext in the late 1980s[1]. Lawyers are interested in the computerisation of a number of different aspects of legal practice, including retrieval of documents relevant to decision-making, other forms of research, the decision-making itself, and the generation of legal documents. We use 'computerisation of law' to encompass both the computerisation of these various aspects of legal practice, and of the legal source materials (such as cases, statutes, commentary) used in them.

Most commercial applications have concentrated on only one of these paradigms. This lack of integration is not peculiar to law, but has been observed to be a general feature of the computerisation of information[2]. There are exceptions, in law and elsewhere, for example integration of hypertext with both free text retrieval[3] and expert systems[4].

The current paradigms have prompted considerable theoretical legal research, concerning such matters as the jurisprudential models implicit in various types of legal expert systems, and the adequacy of Boolean retrieval for legal research. There has been some analysis of the relationship between these different approaches, particularly in relation to the use of AI techniques in information retrieval, or 'conceptual information retrieval'[5], and research on the development of integrated computerised workstations for public administration[6]. Vandenberghe stressed the importance of integration[7], and others have done so since[8]. However, there has been relatively little development of an integrated theory for all aspects of the computerisation of legal materials.

We refer to the comprehensive integration of the modes of computerising law as the 'legal workstation'. This paper describes our approach to such integration, the DataLex Workstation software, and its use in an application to privacy law (the 'Privacy Workstation'). Arguments concerning the practical and theoretical importance of integration are also advanced.

2. DataLex Workstation software

The DataLex Workstation software[9] combines expert systems, hypertext, and free text retrieval into one general-purpose tool. It has been developed for use in commercial applications, and to teach legal applications development, rather than as a pure research vehicle.

2.1. Origins

The Workstation software derives from six separate programs we have developed[10] since 1986: *XSH*, an expert system shell; *AIRS*, a free text retrieval system; *LES*, a text animation package and automated document generator; *HYPE*, a hypertext engine; and PANNDA, an example-based shell for modelling legal precedents. XSH has recently been superseded by YSH, an expert system shell with a quasi-natural language interface. YSH, AIRS and HYPE comprise the three 'engines' within the Workstation software, but with a new common interface.

2.2. Technical features

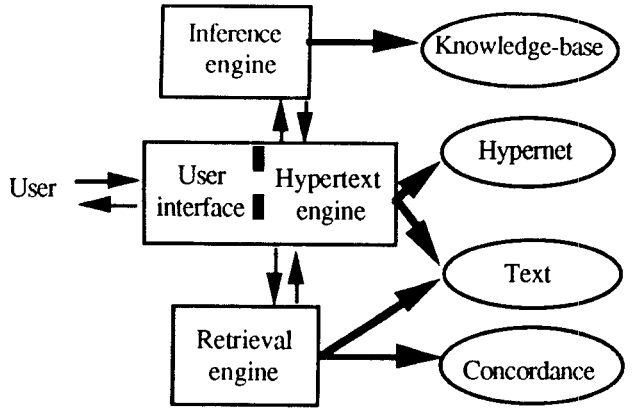
The software is implemented in a portable dialect of 'C'. Originally developed under Unix, it has been ported to DOS. It uses a text-based interface, making it suitable for dial-up telecommunications access. This is desirable because, wherever legal information changes regularly, an on-line application may be required. Despite wide area networks, most users of legal applications are likely to be reliant on slower telecommunications for some time.

2.3. Workstation design and components

The Workstation software incorporates three 'engines' which process legal knowledge and data in different ways: an inference engine, a hypertext engine and a free text retrieval engine. Each communicates with the user through a common user interface (which is based in part on the hypertext engine). We intend to add an automated document generator, for which the YSH inference engine will probably be used, with an event-driven report generator. This addition is

necessary before we can claim that the Workstation integrates the most important current modes of computerising law.

Figure 1: DataLex Workstation architecture



Two elements of integration are that, firstly, the system must appear to the end-user as an integrated whole where all elements interact in a consistent way; and secondly, it should make maximal use of shared knowledge and data. The extent to which these aims have been achieved will be explained in relation to each of the Workstation components.

2.4. Information representations

Each 'engine' requires its own form of representation of the legal 'knowledge' and 'data' which it manipulates: a rule-base for the type of inference engine used in the Workstation; a concordance for text retrieval; and a hypernet (network of nodes and links) for hypertext. Each representation is conceptually distinct from the legal texts which are usually one of its principal sources[11]. The expression 'knowledge representation' is most often used in relation to expert systems, indicating that a knowledge-base involves more than mere data. Concordances and hypernets are more easily seen merely as 'data', but the creation of text retrieval and hypertext systems does involve some addition of legal knowledge to the 'raw' textual sources of the law, such as knowledge of the structure of different types of legal texts. In our view, the distinction between 'knowledge' and 'data' is is often a very fine one. A list of related concepts, for example, is as of much relevance to an expert system inference engine as it is to a hypertext engine (where it can be used as a cross-reference) or a free-text retrieval system (where it can be used to expand search terms). Perhaps it is best to say that the Workstation involves various 'information representations'.

3. Privacy Workstation application

The Australian *Privacy Act 1988* (Cth) and the office of Privacy Commissioner commenced in 1989. Because privacy law is an area of expertise of one of the authors, it presents a good opportunity to build a comprehensive computerised representation of an area of law from its inception. Despite its recent origins, Australian privacy law is developing from a very heterogeneous and complex set of source materials: statutes; regulations; Commissioner's 'guidelines', determinations on exemption applications and determinations of complaints; Court cases; Parliamentary decisions; the Digest of government personal information systems, and academic commentary. It covers a range of closely linked subject matter, including public sector personal records, data matching, spent convictions, tax file numbers and credit reporting. Material from all of these sources is progressively being incorporated into the Privacy Workstation.

All material in the Privacy Workstation is accessible through hypertext and full text retrieval. The conversion of all of the significant statutory sources into expert system components is not yet complete. At present, the system provides advice on the potential applicability of the 11 Information Privacy Principles (IPPs) in the Privacy Act, and on the applicability of the spent convictions legislation. The Privacy Workstation is in commercial use by Australia's largest credit bureaux, is used on-line by the Australian Privacy Foundation (a lobby group), and is being evaluated by the Privacy Commissioner's Office.

4. The hypertext engine

In order to explain their links to the inferencing component, we will first describe the text retrieval and hypertext components of the Workstation software. Hypertext, sometimes called 'non-linear text presentation' or 'text navigation', is defined by Jakob Nielsen[12]:

> Hypertext is non-sequential writing: a directed graph, where each node contains some amount of text or other information. The nodes are connected by directed links. In most hypertext systems, a node may have several out-going links, each of which is then associated with some smaller part of the node called an *anchor*. When a user activates an anchor, they follow the associated link to its destination node, thus *navigating* the hypertext network. Users backtrack by following the links they have used in navigation in the reverse direction. *Landmarks* are nodes which are especially prominent in the network, for example by being directly accessible from many (or all) other nodes.

4.1. Hypertext and law

Legal materials are particularly suited to hyper*text* presentation, whereas other disciplines may be ideal for hyper*media*. Legal texts are densely cross-related, either expressly or by implication: cases interpret statutes; cases cite cases; definitions define terms used elsewhere in the same statute; regulations have as their source the provisions of a statute, and legal commentaries are usually a thicket of cross-references.

It is likely that the hypertext techniques and styles appropriate to law will be particular, at least to some extent. For example, the importance of authoritative definitions (as are provided in statutes) is unlikely to be matched in other domains. Some interesting hypertext applications to law have been reported[13]. Hypertext appropriate to law has yet to be defined, but there is now a considerable literature on the

general features of hypertext and the issues involved in choice of techniques and styles[14].

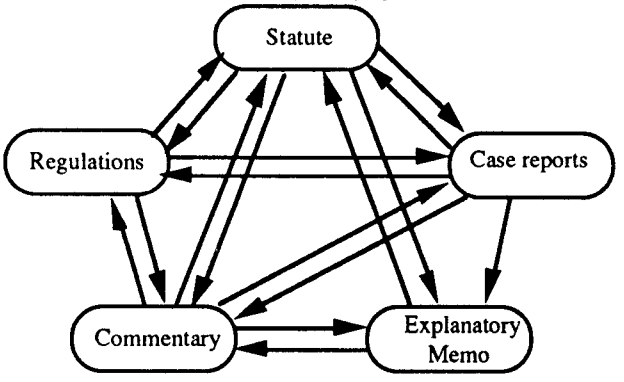
4.2. Hypertext features

Hypertext is implemented in the Workstation software in a way which is intended to be intuitive to users. The main features of the hypertext engine[15], HYPE, from the user's perspective, are as follows. All text navigation is achieved by use of the cursor keypad. The plus key is used to go to a node, and the minus key to return to the previous node. In general, all screens of text, whether containing substantive legal content, menus, indices or otherwise, are treated in the same fashion. Pop-up windows are used for various purposes, but the text within them has the same hypertext features as other text. Any term which is a potential live link ('anchor') appears marked on screen and may be selected by use of cursor keys or by the first alphabetic character of the the term. Text is regarded as being continuous, rather than as discontinuous nodes based on a fixed size screen or 'card'. This allows a user to regard the text as being similar to that contained in a book, by browsing backwards and forwards through the (screen) pages.

Access to any particularly significant node (or 'landmark'), such as tables of contents or indices, is available via a pop-up menu which lists all nodes so designated - the electronic equivalent of the coloured plastic tabs in a loose-leaf service! A list of all nodes navigated by the user (the 'backtrack path') is available to the user as a pop-up list. Any node on the list may be selected to allow quick return to any point in the backtrack path.

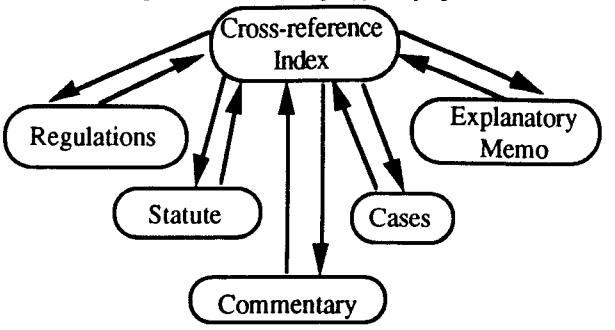
4.3. The cross-reference index

Figure 2: Hyper-spaghetti



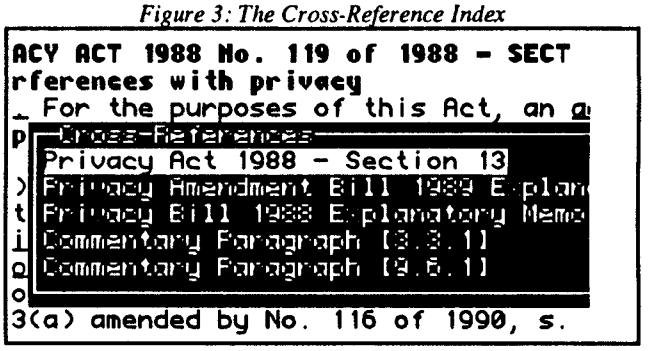
A user who is browsing, say, a section of an Act, should be able to access other textual sources related to that section in an intuitive way. In Australia these include delegated legislation, Parliamentary Explanatory Memoranda, case reports, and academic commentary. Some desired links may be able to be identified from the texts (for example, section numbers) but in most cases where multiple sources are involved, their creation is equivalent to intellectual indexing of hypertext nodes. If we attempt to provide direct links from each of these sources to most other sources (many-to-many links), the resulting hypertext resembles nothing so much as a bowl of spaghetti[16]. Inclusion and maintenance of so many links is near impossible.

Figure 2A: Avoiding hyper-spaghetti



Where a hypertext application is based primarily around a statute, an alternative approach is to eliminate many-to-many links by routing links through a form of table of statutes, which we call a 'cross reference index'[17]. In the Privacy Workstation it contains, for each section of any important Act, a list of links to all other source documents relevant to that section. All links to the cross-reference index become, in effect, bi-directional links to related materials. Each source document contains direct links only to the appropriate section in the index, thereby greatly reducing the number of links which need to be inserted and maintained. These 'cross-reference links' are additional to those which appear explicitly in the source texts. This cross reference index, the list of all nodes which are cross-referenced to the current node being browsed, is accessed in the Workstation as a pop-up window, illustrated in Figure 3.

Figure 3: The Cross-Reference Index



Another approach would be to make all links bi-directional, so that any link leading into a node may be explored, but this often produces far too many insignificant links to be useful. In contrast, the cross-reference index filters out insignificant links.

4.4. Automated hypertext creation

Creation and maintenance of hypertext links in large and complex bodies of text is very difficult. This is particularly so where text undergoes regular change, as is the case with legal commentary, and to a lesser extent with statutes and regulations. If hypertext links are inserted in source documents manually, large or complex hypertext systems become impractical. An early Privacy Workstation, containing less than a megabyte of text (a couple of statutes, explanatory memoranda, and commentary), had over 2,500 nodes to which there were over 8,500 links.

To eliminate manual marking up we create automated marking-up scripts for each category of document which has a reasonably regular form (statutes, regulations, cases, commentary etc). These standard 'templates' can be used to automatically create useful hypertext with almost all of the desired functionality.

However, most documents have non-standard features. For example, while most statutes from common law jurisdictions have explicit interpretation sections in which defined terms are easily recognised, other sections may implicitly define a term ('In this Act, a .... will be taken to be a ...'). These non-standard definitions must be identified and designated as nodes, and the corresponding links identified, to create comprehensive and sophisticated hypertext. Non-standard structural features of documents must also be identified. The *Privacy Act*, for example, provides that each of the eleven Information Privacy Principles (IPPs) contained in s14 shall be interpreted as if it were a separate section (s2). Useful hypertext must treat each IPP as a separate node.

We deal with non-standard structures by modifying the standard marking-up script so that it recognises the non-standard features of a particular document. So, if a document is important enough, it will have its own marking-up script. The alternative is to edit the source texts after the standard automated marking up has been done, but this will be ineffective if the source text is amended. Our method allows all source to documents remain in the raw text form, and the system can be re-created automatically by use of the script files.

5. The text retrieval engine

The text retrieval engine, AIRS, is a conventional free text retrieval system which is in commercial use[18].

5.1. Search features

The retrieval engine relies upon a five level concordance: chapter, article, section, paragraph and word. Plurals and possessives can be (optionally) concorded as being equivalent to their singular forms. Synonyms for specific words and phrases may also be defined, allowing all synonyms of a search term to be retrieved. The retrieval functions are 'exactness', rather than 'nearness', in the sense that retrieved documents are not ranked by some measure of likely relevance to the search request[19].

All search and display commands are invoked by a function key or menu selection. Search terms may be connected by boolean connectors and proximity connectors[20]. A search term commencing with a boolean connector is treated as a search modification and appended to the previous search request. Right hand truncation is allowed. The scope of searches may be limited by a 'search area' option which presents a list of defined subsets of the database which may then be selected or de-selected. The current search is then automatically re-executed over the new search area.

5.2. Display features using hypertext

All displays of search results in the Workstation utilise hypertext. When a search is executed, the list of titles of retrieved articles (the 'retrieved list') is displayed as a hypertext list. Selection of an article from the list takes the user to the hypertext node at the commencement of that article. From that point the user can use all of the hypertext browsing techniques to follow other associations. In that sense, the display of a search result is completely open-ended, in that the user may pursue any associations that a search result suggests. The retrieved list is also treated as a hypertext node, although it has been created dynamically by the search. The other display method is by sequential scanning.

6. The inference engine

The inference engine component, YSH, supports backward chaining, forward chaining and procedural inferencing and uses a quasi-natural language based rule representation[21]. It is integrated in the Workstation so that textual aspects of a consultation (questions, explanations and reports) are presented as hypertext, and consultations can be invoked from hypertext links.

6.1. Inferencing features

YSH rules operate as free-standing pieces of procedural code which are either invoked by user-definable forward or backward chaining or called explicitly from other rules. Rules may be arbitrarily complex and may contain most of the constructs found in conventional procedural languages (for example, assignments and assertions, IF-THEN-ELSE statements and various types of loops).

By default, rules are both backward and forward chaining. The initial decision as to which rule should be evaluated is determined on a backward chaining basis. Whenever an object value is inferred or supplied by the user the system silently invokes all rules where the object is used. If it becomes necessary to request further information from the user, evaluation is halted. The effect is that, at any particular instant, the system will have determined all possible conclusions from known values.

This default behaviour can be altered by specifying that a rule should only participate in backward chaining, forward chaining (with or without user intervention) or neither (procedural only). Regardless of its type, a rule may always be called explicitly from another rule. Any rule may also be called to start a new problem session (i.e any rule may be a 'goal').

6.2. Quasi-natural language rules

YSH uses a fairly conventional parser to read knowledge-bases from standard text files. There is no separate rule editor. Object names are free-form English sentences or phrases and are generally taken verbatim from legislation or other target primary legal materials. Parsing heuristics automatically effect the necessary transformations of these names so that the system can generate questions, translations and explanations, as described below. If inappropriate parsing occurs, these rules can be modified by the application developer (for example, by specifying that a word is a verb or by dividing a statement up into a subject and predicate) or

overridden if necessary. Such intervention is usually not necessary.

This approach, whilst not new[22], is worthwhile in that it promotes transparency of the knowledge-base and makes the implementation and maintenance of large and complex rule-bases easier. Knowledge representations which separate rule-bases from object descriptions governing user interaction (such as prompts, translations and explanations) have been correctly criticised by Johnson and Mead[23] on the grounds that maintenance of each parallel component of the knowledge representation introduces further problems of maintenance, potential unreliability and cost.

Subjects which form part of object descriptors (such as *the information* in the example below) can be named dynamically. The translation facility will automatically prompt for a name (and gender if necessary) and use this and/or appropriate pronouns and possessives in subsequent questions, translations and explanations. Information gained about subjects in this way is also available for general use (for example, the fact that a subject is a natural person, rather than a company, will often be relevant, as may be a natural person's gender).

The overall rule syntax also fits into a quasi-natural language mold. In many cases, there is more than one way in which the same basic construct may be expressed, so that rules can be made to approximate the wording of legislation and other legal materials.

Figure 4: Information Privacy Principle 1/1

```
1.  Personal information shall not be collected by a
collector for inclusion in a record or in a generally
available publication unless:

  (a)  the information is collected for a purpose that is
a lawful purpose directly related to a function or
activity of the collector; and
  (b)  the collection of the information is necessary for
or directly related to that purpose.
```

Figure 5: The equivalent rule for IPP1/1

```
THING the information
PERSONTHING the collector

RULE Information Privacy Principle 1(1) PROVIDES
the prerequisites of IPP 1(1) are met ONLY IF
  the information is personal information AND
  the information was collected by a collector AND
  the information was collected for inclusion in a record
or in a generally available publication AND
  the exception provided in IPP 1(1) (a) and (b) does not
apply

RULE IPP 1(1) (a) and (b) PROVIDES
the exception provided in IPP 1(1) (a) and (b) applies
ONLY IF
  the information was collected for a purpose that is a
lawful purpose directly related to a function or activity
of the collector AND
  the collection of the information is necessary for or
directly related to that lawful purpose
```

6.3. Explanation and reporting facilities

YSH provides six types of explanations of the inferencing process and its results: (i) 'Facts', a list of user-supplied object values; (ii) 'Conclusions', a list of inferred object values; (iii) 'How', an explanation of how a particular conclusion was reached; (iv) 'Why', an explanation of why a question is being asked; (v) 'Rule', a display of the rules being evaluated; and (vi) a final 'Report' explaining how the final conclusions concerning the consultation goals were inferred.

The following simple example gives the general flavour of a YSH dialogue, but does not show the Workstation interface, such as the availability of hypertext links from prompts.

Figure 6: A small consultation on IPP1/1

```
 1)   What is the name of the information ?
   ** Smith's Medical Record
 2)   Is it personal information ?
   ** Yes
 3)   Was it collected by a collector ?
   ** Yes
 4)   Was it collected for inclusion in a record or in a
generally available publication ?
   ** Yes
 5)   What is the name of the collector ?
   ** Medicare
 6)   Is the collector a natural person ?
   ** No
 7)   Was Smith's Medical Record collected for a purpose
that is a lawful purpose directly related to a function or
activity of Medicare ?
   ** Why
  This will help determine whether or not the exception
provided in IPP 1(1) (a) and (b) applies.

 7)   Was Smith's Medical Record collected for a purpose
that is a lawful purpose directly related to a function or
activity of Medicare ?
   ** Yes
 8)   Is the collection of the information necessary for
or directly related to that lawful purpose ?
   ** Yes

      REPORT
FACTS
  Smith's Medical Record is personal information. It was
collected by a collector. It was collected for inclusion
in a record or in a generally available publication. It
was collected for a purpose that is a lawful purpose
directly related to a function or activity of Medicare.
The collection of the information is necessary for or
directly related to that lawful purpose.

DECISION
  The prerequisites of IPP 1(1) are not met.

REASONS
  The prerequisites of IPP 1(1) are not met because the
exception provided in IPP 1(1) (a) and (b) applies. The
exception provided in IPP 1(1) (a) and (b) applies because
Smith's Medical Record was collected for a purpose that is
a lawful purpose directly related to a function or
activity of Medicare and the collection of the information
is necessary for or directly related to that lawful
purpose.
```

All of these explanation and reporting mechanisms are object-driven rather than being event-driven. As object values are determined, the system records the reasons for the decision based upon which objects were used to arrive at the result. These reasons can be used to explain why a question is being asked, how a result was arrived at or in an overall report which appears at the end of the problem session. An attempt is made to record only significant information, so that the simplest and most straight-forward explanation of any given result can be provided.

During a consultation, the user may also instruct the system to 'forget' a previous user-supplied value, causing conclusions relying on that value to be re-inferred.

6.4. Hypertext links from the inference engine

In addition to the explanation facilities provided by YSH, the Workstation provides a different form of 'explanation' through hypertext access from YSH-generated explanations to the legal source materials on which those explanations are based, and to which they refer. Text generated by YSH (questions, explanations and reports) is presented to the user as hypertext in which terms appear marked as links in the usual hypertext fashion. Any term, including express or implied definitions in an Act, or section numbers, may constitute a link. If the user selects such a term from a YSH prompt or explanation window, the system takes the user to the hypertext node to which that term is linked, and all normal hypertext browsing can proceed from there. The user can backtrack to the YSH window in the usual way.

In addition to hypertext access from a consultation, the text retrieval engine may be invoked by the user at any time during a YSH consultation, so that free text searches may be conducted in relation to terms used in prompts or explanations. This does not require any explicit links between the search and inferencing engines.

6.5. Hypertext access to the inference engine

A method of integration which operates in the reverse direction is provided via the cross-reference index. If a rule in the YSH knowledge representation is designated as a goal, it will appear in the relevant cross-reference indices. For example, in the Privacy Workstation, the applicability of each of the Information Privacy Principles is a goal which may be inferred separately, so if the user is browsing any part of the hypertext which invokes the cross-reference for IPP 11, Privacy Act s14, then one of the 'cross-references' that will appear is 'Goal: applicability of IPP 11'. If this is selected, YSH will be invoked and the applicability of IPP 11 inferred. This is one method by which hypertext may provide a 'front end' to an inferencing engine, discussed below.

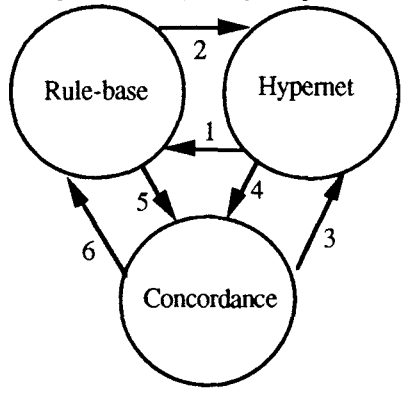
7. The significance of integration

Having described the operation of the Workstation and the integration of its components, it is now appropriate to consider whether this type of integration is of practical or theoretical significance.

7.1. Varieties of integration

The three representations (Rule-base, Hypernet and Concordance) utilised by the three engines present six possible types of of integration, in the sense of access from one representation to another, as described below.

Figure 7: Six types of integration



(1) *Hypernet to rule-base:* A hypertext node may invoke the inference engine to evaluate a goal associated with that hypertext node. This can be considered as a distributed expert system[24]. The Workstation allows user-controlled hypertext access to goals to be evaluated via the cross-reference index (see 6.5), but automatic evaluation of a goal could also be triggered by a hypertext node.

(2) *Rule-base to hypernet:* The Workstation's uses of hypertext terms in its explanation facilities (see 6.4) is an example of an 'open-ended' expert system, discussed in detail in 7.3.

(3) *Concordance to hypernet:* The hypertext display of search results in the Workstation exemplifies this integration (see 5.2). In effect, a search is retrieving and displaying nodes in the hypernet. Because the hypertext representation of retrieved articles may suggest to the user other associated articles which are not in the retrieved list, hypertext display can be seen as increasing the effectiveness of the search[25].

(4) *Hypernet to concordance:* The Workstation allows the user to conduct a free text search at any time while traversing the hypernet, a low level of integration. A node may also trigger a pre-determined automated search, providing, in effect, a context-sensitive interface to text retrieval. For example, if the Commentary on the IPPs in the Privacy Workstation is to be kept up to date, it must refer to any exemption proceedings relevant to a particular IPP. It may be easier to update databases of source materials more regularly than it is possible to revise commentary. One way of keeping the Commentary on, say, IPP 11, up to date would be to place a link referring to recent exemptions in the commentary on IPP 11, such that when the user selected it, an automated search for all exemption proceedings dealing with IPP11 would be commenced. This would be a 'context sensitive' (rather than 'intelligent') front end for information retrieval.

(5) *Rule-base to concordance:* Similar to what is said in (4) above, the Workstation allows free text searches during an expert system consultation, but a rule could also trigger an automatic free text search in similar circumstances. Such a

search could have variable content depending on values of variables obtained during the consultation. The potential of expert systems as intelligent front-ends for text retrieval is well known[26].

(6) *Concordance to rule-base:* Free text searches could also retrieve any rules in the rule-base which contain the desired search terms. Such searches over the rule-base are meaningful because of YSH's quasi natural language knowledge representation. Such searches might be informative in themselves (at least for the expert user), but would be far more valuable if any goal rules so retrieved were then able to be evaluated. This can be implemented in the Workstation by use of the cross-reference index.

This sketch of the variety of integrations made possible by the Workstation indicates its advantages: it allows a wide variety of different types of 'real world' applications to be built (see 7.4); and it provides a platform for some interesting theoretical research (see 9.2).

7.2. Enquiries across three dimensions

The six types of integration sketched above raise a more general theoretical question: how should we conceptualise and describe what a user is doing in using an integrated application such as the Privacy Workstation? A user who wishes to solve a legal problem may start her use of the Workstation by an initial menu selection which invokes any one of the three engines and its information representation. In the course of solving the problem, use of the system may take the user 'out' to one or both of the other information representations, and possibly (but not necessarily) back to the one she started with, until at some point the user feels that she has extracted as much useful information on the problem as the Workstation is capable of providing.

The terminology to best describe use of such systems is uncertain. 'Search' is associated with free text searching, 'browse' with hypertext, and 'consult' with expert systems. In the absence of a better term, we describe use of the Workstation as an 'enquiry'. An enquiry is conducted across three representations using three engines to manipulate information in different ways. We could refer to overall use of the Workstation as an 'enquiry across three dimensions'.

To describe the operation of such applications from a user's perspective, it may be useful to generalise the notion of a hypertext 'path' or 'thread' to at least include the rule-base, if not the concordance. In the Workstation, only some rules are goals, which means that they are access points from the hypertext, but any rules can provide exit points to the hypertext (and may be returned to). Although it is incorrect to think of a rule-base consultation as involving a sequential path between rules (like a path between nodes), this may often be how use of the rule-base appears to a user, with accesses to and exits from particular rules. Searches, on the other hand, cannot be regarded usefully as involving any sequential path through the concordance. A search is usually a single event (although searches may be refined or replaced by successive searches) after which the user goes back to the hypertext to display the result, or to the consultation if the result is to be evaluated in some way.

What we are trying to avoid is a conceptualisation which makes any one of the three engines / representations central and the others mere appendages. The Workstation is not 'really' an expert system with some useful explanation mechanisms tacked on (see 7.3 below). Nor is it best regarded merely as a hypernet in which small rule-bases and automated consultations are embedded[27]. Neither approach does justice to the flexibility with which users may commence, conduct and conclude an enquiry using any of the representations.

7.3. Open-ended expert systems

Should we aim to automate legal decision-making, or to provide support for legal decision-makers? One of the most difficult problem in the development of legal expert systems is that caused by the open texture of legal language. The problem arises in a number of ways[28], and may be resolved by one or more of three possible reasoning agents: the inferencing engine; the knowledge-base developer; or the end user. Differences over the most appropriate response to this problem leads to different models for the development of legal expert systems. One approach is to develop software and knowledge representations which will suggest solutions to open texture issues[29]. Such research presents challenging theoretical problems, and is as yet difficult to implement in commercial systems.

Another approach, concentrating on supporting and augmenting the decision-making abilities of the system user to resolve open texture problems, has been advanced by a number of researchers[30], one of whom sees 'less need for a cognitive legal machine than for a less sophisticated but more humble product to support intelligent human interaction' [31]. Taking this approach, it becomes crucial for legal expert systems to allow access to as rich a collection of support materials as possible, so as to support intelligent choices by the user when interpretation of an 'open textured' predicate is required (ie when the user has to make a choice which the system is incapable of making). The user needs to be given open-ended access to the relevant supporting materials, rather than for the system to simply direct the user to a few definitions which may assist interpretation ('closed-ended' assistance).

We take this second approach in the Workstation, by providing various methods by which a user may move from an expert systems consultation to the interpretive materials relevant to that point in the consultation (for example, a statutory definition of a term used in a prompt). Because these relevant materials are presented as hypertext, the interpretive process is open-ended, with the user able to pursue associations, or to conduct free text searches, until the interpretive resources of the system are exhausted. One advantage of this approach is that it may help overcome some aspects of the 'brittleness' of expert systems: a user can use the other resources of the system to help 'work around' factual variations not adequately dealt with by the expert system itself.

A useful way to view a legal expert system, from the perspective of the user, may be as an interaction between a semi-expert inferencing system and a semi-expert user/interpreter, with control over the course of the problem's solution alternating between the two parties to the

interaction. Each does what (s)he or it does best, then hands back control to the other. The program controls those steps in the solution process that are capable of being embodied in a computerised inferencing agent, given existing technology. The user controls those steps of the solution process which involve abilities which cannot (at least as yet) be so embodied, including the lawyer's various interpretive skills.

Susskind's rather pessimistic conclusion that research should concentrate 'on designing systems to solve clear and deductive cases'[32] might be overcome by making the enhancement of the interpretive resources of users a central aim of legal expert systems research. A key practical question may be to find the boundary between those elements of open texture problems that legal expert systems can handle (given existing technology) and those elements that users must provide. Research into non-deductive methods of inferencing may, over time, push back this boundary.

7.4. Practical advantages

The main value of integrated tools in the building of 'real world' applications is that they can save application developers from attempting to use the techniques of one mode of computerisation for purposes for which it is not suited. Attempting to force square pegs into round holes is rarely satisfactory. It is sometimes difficult to anticipate at the outset of a project what combination of tools will be needed. Successful application development is aided by the availability of as wide as possible a choice of tools, and the ability to mix their use in ways which are easy to develop and transparent to the user.

8. Terminology and theory for integration

In writing this paper we have felt the need for a more adequate conceptual framework, and an accepted terminology, to describe the integrated legal computerisation which the Workstation attempts to advance. We have used ad hoc terminology such as 'computerisation of law', 'information representation', 'enquiry in three dimensions' and 'cross-reference index' in attempting to overcome this.

There is a need for a consistent and comprehensive conceptual framework and terminology describing all of the elements of computerisation of legal information, incorporating the existing paradigms, and identifying where law's needs are distinct from those of other domains. We could call this a 'grammar'[33] for the computerisation of law, for want of a better term to encompass both conceptual framework and terminology. An analogy[34] to the development of the cinema has been used to illustrate the idea of a hypertext as a new grammar for text. The potential of the new technology of film was at first little understood, to the extent that cameras were bolted down and all shots taken from a central, fixed, perspective. It took many years for film-makers to develop the 'grammar of motion pictures'[35] with which both viewers and film-makers are familiar today, through such contributing techniques as moving cameras, close-ups, flashbacks (a non-linear technique), aerial perspectives and slow motion.

Such a grammar needs to be understood both by application developers, so that they may learn from each other's work, and by the 'audience', the users of computerised legal products. Film audiences may have been confused by film sequences out of their correct temporal order until the concept of the 'flashback' became one that they understood and were comfortable with. A founder of hypertext states 'we have to invent other document forms that somehow become standard so that people have pattern recognition and say "Ah, yes, I know how that one works"[36]. Styles of application development appropriate for law will give us that recognition in the legal domain.

A starting point is to distinguish at least three levels of description. The highest is a 'mode' or paradigm approach to computerisation, such as expert systems, hypertext, etc. The second is that each mode has computing 'techniques' characteristically associated with it, such as inferencing by backward chaining rules (in expert systems), backtracking (in hypertext), or word occurrence proximity searching (in text retrieval). The third level,'styles' of computerisation, is the most important for the development of a grammar characteristic of law.

Most computing modes and techniques are not peculiar to law, but there may be characteristic styles in which they are used to produce effective legal applications. For example, the ideal of logic programming that 'every rule in such a system is supposed to be a formal paraphrase of some clause in the legislation'[37] is not dictated by Prolog, but derives from arguments about what makes a legal expert system legally sound. Similarly, hypertext facilitates but does not dictate cross-referencing structures such as we and others have used, but some form of such structures may be needed for effective hypertext applications based on legislation. The ways in which available techniques are combined by application developers into applications that are powerful but intuitively easy to use is a matter of style, distinguishing good applications from bad ones. The literature on legal computerisation has concentrated on new techniques, but we also need a literature on good style in legal applications development.

9. Future development and research

9.1. Workstation enhancements

We do not consider any of the three engines on which the Workstation software is based to be complete. Each could be usefully enhanced. For example, the text retrieval engine would benefit, at least for some applications, from the addition of relevance ranking of retrieved articles. A more 'interventionist' search interface[38] is also desirable. The hypertext engine could also usefully support user-initiated links and nodes for some 'workgroup' applications. As mentioned, the inference engine needs modifications to deal with document generation. Facilities by which text generated by the different engines may be combined into reports and other documents in a user-controlled fashion are also needed to maximise the Workstation's potential as a practical tool.

For research purposes, more significant enhancements to the Workstation will come from integrating with its existing engines and interface some less conventional methods of

inferencing, searching or browsing, such as inferencing engines which are not rule-based. For example, we may incorporate a modified version of PANNDA (precedent analysis by nearest neighbour discriminant analysis), which relies on statistical techniques to handle case-law reasoning[39]. Like many methods for computerisation of non-deductive reasoning, PANNDA requires expert analysis of a set of cases to determine the attributes which are of legal significance. PANNDA was incorporated in XSH so that the same attributes and their values may be used in rule-based inferencing and 'precedents' (PANNDA attribute sets), and the same can be done with YSH.

9.2. Future research possibilities

The integration of tools in the Workstation, even at its present level, should make it possible to conduct some interesting future research, initially by investigating how far the six approaches to integration sketched in 7.1 can be developed to produce new and useful methods of 'enquiry'.

For example, a version of the 'norm based thesaurus' described by Bing[40] could be implemented by using the Workstation, because of two factors: YSH's use of English words and phrases as object names; and the integration of the Workstation components. Each object in a YSH rule-base would require a synonym list. A user's inference engine consultation would lead the user to a relevant rule (using the 'Rule' explanation facility), the equivalent to Bing's 'node'. A free text search for an object or objects named in that rule would then, by use of the AIRS synonym facility, retrieve all documents relevant to that rule, irrespective of the terms used therein. The reverse direction of integration, as described by Bing, would be achieved simply by treating each rule in the rule-base as a separate article for free text retrieval purposes, and by cross-referencing each rule to its relevant statutory source using the cross-reference index. A search for any related terms would then retrieve the relevant rules, because of the synonym lists, and any goal rules retrieved could then be used to commence a consultation via the cross-reference index. Our version differs from Bing's in using an inference engine rather than a graphical representation of the normative structure.

Addition of a document generator or other inferencing methods will increase the number of different types of integration which are open for exploration.

Bing has also raised 'the possibility of building a parser which will use the statutory text as input, and have an approximation of [an expert system in] normalised form as output'[41]. We have largely automated the text handling for the creation of hypertext and free text retrieval for some types of texts, particularly legislation. YSH already has a quasi natural language representation, and parsing heuristics to manipulate it for reporting purposes. It would therefore be consistent with our current work (but much more difficult) to investigate the extent to which parsing heuristics can automatically convert legislative text into a YSH rule-base[42].

However, the main aim behind the YSH knowledge representation was to make it relatively easy for a person, not a program, to create non-trivial knowledge-bases from legislation, by little more than legally skilled paraphrasing. Such knowledge-bases will not, in themselves, deal adequately with problems of open texture and logical ambiguity. We intend to explore whether such knowledge bases become more valuable, despite the relatively low expertise that they embody, when they are integrated with hypertext and free-text representations of the same sources. If so, commercial production of useful legislative expert systems will be advanced.

The present Workstation only provides for hypernet access to commence evaluation of those rules in the rulebase which are designated as goals, and in that sense access to the rulebase is through a series of pre-defined 'corridors'[43]. It may be much more valuable for a user to be able trigger the evaluation of the corresponding rule(s) from any section of the legislation in the hypernet. Such integration would put an expert system's behaviour more within the user's control.

9.3. Integration as a way ahead

Leading scholars of both legal text retrieval and expert systems have suggested that research and development have not advanced very far in the past decade[44]. One way forward, we suggest, is to give greater recognition to the importance of integration of the existing approaches to computerising legal information, at both the theoretical and practical levels.

---

[1] For brief histories: on text retrieval see G Greenleaf, A Mowbray & D Lewis *Australasian Computerised Legal Information Handbook*, Butterworths, 1988, Chapter 4, or more comprehensively, J Bing (Ed) *Handbook of Legal Information Retrieval* North Holland 1984 Part III; on expert systems see A Tyree *Expert Systems in Law* Prentice Hall 1989 Ch 1; on document generators see S M Brooks *Computerizing for Personal Productivity* Butterworths, Toronto, 1989, Chapter 5; on hypertext see J Smith and S Weiss 'Hypertext' (1988) 31(7) *Communications of the ACM* p816.

[2] H P Frei 'The Future of Information Systems', Seminar paper, CIRCIT, Melbourne, 1990.

[3] For example, DiskROM Australia uses the *Innerview* software which combines free text retrieval and hypertext techniques.

[4] For example, P Johnson and D Mead's *STATUTE* software (Softlaw, Canberra) provides hypertext access from terms which arise in an expert systems dialogue.

[5] See the papers by Tong et al, Hafner, Bing, Dick and Belew in *Proc. 1st ICAIL* 1987, and by Rose and Belew in *Proc. 2nd ICAIL* 1989

[6] See the papers presented at the *7th Colloquy on the use of computers in administration of justice - Integrated work stations in the legal sector and decision support systems*, Council of Europe, Lisbon, 1988; see also J Bing 'The concept and design of integrated work stations for public administration', International Council for IT in Government Administration, 22nd Conference, Estoril, 1988

[7] G Vandenberghe 'Software oracles' in H W K Kaspersen and A Oskamp (Eds) *Among Friends in Computers and Law* Kluwer, Deventer, 1990; see also A W Koers et al 'Delphi

revisited: The mythology of the lawyer's electronic workbench' in the same volume.

[8] A Oskamp and P van der Berg 'Legal expert systems and legal text retrieval systems: how about integration?' in Kaspersen and Oskamp, op cit

[9] DataLex is the name used by the authors for their joint research since 1985. The research presented here has been assisted by a grant from the Australian Research Council for the development of the Privacy Workstation. The Workstation software was designed by Andrew Mowbray and Graham Greenleaf, and implemented by Andrew Mowbray

[10] All of the programs are by Andrew Mowbray, except PANNDA by Alan Tyree.

[11] See J Bing 'Rules and representations' in P Blume (Ed) *Nordic Studies in Information Technology and Law*, Kluwer, Deventer, 1991.

[12] J Nielsen 'The art of navigating' *Communications of the ACM* (1990) Vol 33 No 3, p298

[13] For examples, see C Stephen and H Schreiber 'CD-ROM, hypertext and the law', Australian Law Librarians Conference, 1989; E Wilson 'JUSTUS: A workstation for information retrieval in law' *Informatica e Regolamentazioni Giuridiche*, Rome 1988; D Painter 'Hyperlaw', *Law Technology Centre & Bileta Newsletter*, Vol 2 No 3, 1990.

[14] See the papers in the Hypertext Special Issue (papers presented at Hypertext '87) 31(7) *Communications of the ACM*, July 1988; E Barrett (Ed) *The Society of Text – hypertext, hypermedia and the social construction of information*, MIT Press 1989; R McAleese and C Green *Hypertext – state of the art*, Ablex, NJ, 1990.

[15] See G Greenleaf & A Mowbray *The Privacy Workstation User Manual*, April 1991.

[16] '...in a sense hypertext gives us a goto, and a goto, as we all know, produces spaghetti...': A van Dam, 'Hypertext '87 Keynote Address' (1988) 31(7) *Communications of the ACM* p891

[17] In DiskROM Australia's *Corporations Law* CD-ROM, a similar device is called an ACTRIX.

[18] The AIRS search language is a superset of that of STATUS, which it was originally developed to emulate. AIRS is used by LINK (the Lawyer's Information Network) to run ESTOPL (a case abstract database), and by Monash Law Library for ALLI (Australian Legal Literature Index).

[19] See Jon Bing 'Performance of text retrieval systems: the curse of Boole' (1987) 79 *Law Lib J.*

[20] Boolean: AND, OR, NOT; Proximity: NEAR, for same paragraph; /n,m/, for specified word proximity; and WITHIN for within a named section of an article

[21] A Mowbray *YSH Reference Manual*, DataLex 1991

[22] See the work of D Waterman et al on ROSIE; for example D Waterman, J Paul and M Peterson 'Expert systems for legal decision making' *Proc. 2nd Aust. Conf. on Expert Systems* NSW Institute of Technology, 1986 and D Waterman and M Peterson 'Models of legal decision making' in P Clahr and D Waterman (Eds) *Expert Systems - Techniques, tools and Applications* Addison Wesley 1986 Ch 5

[23] P Johnson and D Mead 'Legislative expert systems', (unpublished) Softlaw Corporation, Canberra, 1989; and 'Natural language - An appropriate knowledge representation scheme for legislative expert systems' (unpublished) Softlaw Corporation, Canberra, 1991. Their approach is implemented in their STATUTE software.

[24] P A Carlson 'Hypertext and intelligent interfaces for text retrieval', p71 in E Barrett (Ed) *The Society of Text* MIT Press 1989

[25] cf P A Carlson *op cit* p65

[26] See Oskamp and van der Berg *op cit* for a legal example.

[27] cf P A Carlson *op cit* p71

[28] See R Susskind *Expert Systems in Law* Clarendon Press, Oxford, 1987 Ch 5

[29] For different approaches see G-J van Opdorp and R F Walker 'A neural network approach to open texture' in Kaspersen and Oskamp *op cit*; E L Rissland and D B Skalak 'Interpeting statutory predicates' *Proc. 2nd ICAIL*, 1989 p46

[30] See Taylor and Brown 'Supporting local office adjudication' and Taylor 'The DHSS local office demonstrator' (both papers Alvey DHSS Large Demonstrator Project, Dept. of Systems, University of Lancaster, 1988); A Berg et al 'Developing a KBS support system for handling social assistance', SAFAD Stockholm, 1988; and P Johnson and D Mead *op cit*

[31] R Stamper, book review, *The Times* 1988

[32] Susskind *op cit* p192

[33] 'grammar n. 1. the features of a language ... considered systematically as a whole, esp. with reference to their mutual contrasts and relations... 4. speech or writing in accordance with standard usage... 5. the elements of any science, art or subject...' (*Macquarie Dictionary*)

[34] C Stephen and H Schreiber 'CD ROM., Hypertext and the Law' *op cit*, quoting J Anderson 'Interactive multimedia: discovery by design' *MacUser*, March 1989 pgs97-8

[35] J Anderson *ibid*

[36] A van Dam *op cit*

[37] T Bench-Capon 'Deep models, normative reasoning and legal expert systems' *Proc. 2nd ICAIL* 1989, p37

[38] J Bing 'The text retrieval system as a convers[at]ion partner' 2 *Yearbook of Law, Computers and Technology* Butterworths (UK) 1986

[39] A Tyree *op cit* Chs 7 and 8

[40] J Bing 'Designing text retrieval systems for "conceptual searching" ' *Proc. 1st ICAIL* p43

[41] J Bing *ibid* p47

[42] See the work of L Allen and C Saxon, for example 'Some problems in designing expert systems to aid legal reasoning' *Proc. 1st ICAIL* 1987, p94, and 'Computer-aided normalising and unpacking: some interesting machine-processable transformations of legal rules' in C Walter (Ed) *Computer Power and Legal Reasoning* West, 1985, p495, for the difficulties inherent in such an approach.

[43] See A Galtung and D S Maesel 'Xcite' *Proc. 2nd ICAIL* 1989 p81, contrasing]corridor' and 'marketplace' models.

[44] See J Bing 'Performance of text retrieval systems: the curse of Boole' (1987) 79 *Law Lib. J.*. and L T McCarty 'Artificial intelligence and law: How to get there from here' *Ratio Juris* Vol 3 No 2 1990, p189.