Information Technology and Legal Education: Towards 2000

9th & 10th April 1992
Sponsored by
Context Ltd; Linklaters & Paines; Needham & James

Reducing the Babel of Lawyers

Eve Wilson

Computing Laboratory, University of Kent at Canterbury

Keywords: information retrieval, hypertext, hybrid systems

1. Introduction

Computing in law is increasingly a Babel of competing systems: only polyglots can begin to communicate. Not only does each system have its own voice, but the competing languages have no common philosophy, and there is little hope that the systems will ever talk to one another. Nor is this entirely the fault of competing commercial interests intent on locking customers into a single familiar system. The legal profession itself is spawning a cacophony of expert systems, computer assisted learning packages and litigation support systems that have been hand-crafted for a specific computer milieu. Not only are the working systems not portable, all too frequently even the data they use must be completelyre- structured before it can be exploited in a different environment. Consequently, the same data often exists in several different forms, the form depending on the application. Thus, the same data can have many incarnations: for electronic publishing, for information retrieval, for expert systems and for computer-assisted learning. This is profligate. Instead of only one database to update there may be three or four. This may severely shorten the useful life of a system - particularly if the original implementor is not available to graft amendments into his package. It also introduces time delays in system updates and increases the likelihood of error and inconsistency. Finding common standards for the transfer of information between systems should be an urgent concern of the legal profession.

One way to achieve this is by using data formatted with a standardised mark-up language as the starting point for all applications. This does not mean that the format cannot be altered within the application -although if it need not be, that is a saving. However, it should be possible to make the required structural changes automatically and to reverse the process at will. For example, some data in standard format is used by hypertext package A. It is converted into the structure required by A and, within the system, it is amended, enlarged, given enhanced features. If, at a later date, the same information is required by hypertext package B, then it is re-converted from A's format to standard format and then into the form required by B. Nothing is lost; no work is wasted.

Putative solutions do exist; what is needed is a greater awareness and will to ameliorate the existing confusion and dissonance. Below we look first at the scale of the problem and then consider how to

resolve some of the more pressing difficulties.

2. The Legal Environment

2.1 Law databases

The database of legal and related documents is voluminous and dynamic. Sarah Nichols, in Law Databases 1991, [Nichols 91] lists over 100 databases, some international, some European and some national from fifteen different countries. For European law, she lists 12 databases involving more than eleven hosts: see figure 1.

Database	Host	Content
ABEL	EUR-OP	Official Journal of EC
CELEX	Eurobases	Official database of EC law
CBNC	Dialog Information Systems Pergamon Financial Data Services Data-Star Reuter Textline	Information on Chemical Industry worldwide, with emphasis on Europe.
ECLAS	Europbases	Bibliographic Database of EC
EUROSCOPE	Profile Information Services (UK host) Infotrade (Belgian host)	Policy and legislative developments
INFO92	Eurobases	Legislative progress towards 1992
JUSletter	ECHO	Concise information on European legislation
JUSTIS	Contect Ltd	For EC, Community Law, law reports directives and regulations
LEXIS	Mead Data Central	European law & law reports
RAPID	Eurobases	Press relese material
SCAD	Eurobases	Bibliographic database for EC
SPEARHEAD	Profile information Services Jusis etc	Current & prospective EC measures

Figure 1: European Law Databases

For United Kingdom law Nichols lists 16 databases, involving some 15 different host systems: see figure 2. There are two main problems:

i) There are many different information retrieval system hosts. They perform essentially the same task but the syntax of the access commands vary from system to system and the internal tagging needed in the database differs between systems.

ii) Even within a single host system, related information may be held in distinct databanks or files with no direct communication between the two.

Database	Host	Content
ACOMPLINE	ESA/IRS	Abstacts of books, reports, etc. on social policy

		and urban affairs
British Official Publications (HMSO)	Dialog Blaise-line	Official publications of HMSO
BRIX	ESA/IRS Dialetch	Research Materials relating to the construction industry
Chemical Business Newsbase (CBNC)	Dialog Datastar PFDS	Information on Chemical Industry worldwide with emphasis on Europe
Chemical Safety Newsbase (CSNB)	Dialog Orbit	Hazardous effects of chemical processes with information on legislation & standards
Criminal Justice Periodical Index	Dialog	100 journals, law reports etc. on criminology in US, Britain & Canada
Estuarine Law Bibliography	Brynmore Jones Library via JANET	Estuarine & Coastal Law
ICONDA Periodicals	STN International ORBIT Search Service	Building construction architecture and town planning
Index to legal periodicals	H W Wilson Company	
Lawtel	Prestel	English Case and Statute law
LEXIS	Mead Data Central	Full text database of cases & statutory material for England, Scotland, Ireland, etc.
POLIS	UCC Systems Management Ltd	Parliamentary questions, proceedings & publications
Prestel	British Telecom	Legal information & advice
STARS	EPRC	Legislation, rules and regulations affecting business in Britain
URBALINE	ESA/IRS	Law reports, press comment, social policy & urban affairs
WATERLOWS	PDFS	Directory of solicitors

Figure 2: UK Law Databases

2.2 Tasks within the legal domain

- Information retrieval

The use of databases to retrieve information was the earliest computer application in law, but its growth has been slow. (For an account of information retrieval in law, see [Bing 80,86] and [Campbell 84]) Even working in their native language within their national legal system, many lawyers have been reluctant to embrace computerised information retrieval. Reasons for this are not clear; while financial considerations may be partly responsible, informal research suggests that the user interface has been the major deterrent. Information retrieval techniques for full text have been static for twenty years. Lawyers need training in the use of concordances and in the formulation of boolean queries. Many find this alien to their normal mode of thought and operation.
In addition, lawyers increasingly need information about legal systems with which they are not familiar - information that may be available only in a foreign language from systems which use different access procedures and different query languages to accomplish the same legal task. There are many tasks in addition to information retrieval, and these have developed their own philosophies and interfaces.

- Document drafting

Document drafting is usually divorced from information retrieval. In most firms it is done on a word

processor, perhaps from a machine readable precedent created in the same office. However, documents are still exchanged in hard-copy format only. Some larger firms now have scanners for optical character recognition. More usually, if a solicitor wishes to amend a contract sent for inspection it must be re-typed; different versions of a document are compared by eye; the process of finding and checking alterations is recursive and inefficient.

- Expert systems

Expert systems are primarily of use to a lawyer working in a field that is unfamiliar so that he may quickly check whether there is a case to answer. They have been implemented using almost every programming tool available; there is no standardisation. Consequently, there are almost as many languages and dialogue techniques as there are systems: Cobol, [de Feu 80]; Prolog1 [Sergot 86]; Apes, [Hammond 86]; Crystal II, [Capper 88], LES, [Mowbray 85).

- Computer Assisted Learning

Computer Assisted Learning (CAL) packages are related to, and sometimes indistinguishable from, expert systems; hence, they share the same vices and virtues. Interesting work is being done: HiDES, [Colson 90]; CIJA and FAMKB, [Edwards 90] and Hyperlaw, [Painter 90], and the CMLCAL project with Lexical [Leith 91] is a timely endeavour to foster standards and co-operation. However, lack of resources to create CAL tutorials - and keep them up-to-date - is an unresolved problem. For a fuller appreciation of technology (and the lack of it) in legal education, see [Paliwala 91a, 91b], and the report of the BILETA Inquiry, headed by Professor Bernard Jackson, into the provision of information technology in UK law schools, [BILETA 91].

- Litigation support

Litigation support began tentatively with the use of relational databases to store references and indexes to paper documents held in filing cabinets; imaging has given impetus to computerisation, [Matthews 91]. In the United States similar techniques for the presentation of evidence have already spread to the courts, [Bondi 91]. Complex cases may involve keeping track of hundreds of different documents with annotations and opinions from diverse sources relating to different points of law. Work in this field is still in its infancy, but some firms are now using hypertext to link disparate pieces of information: [Yoder 89]. Management is the real problem and this, together with the proper integration of hypertext systems with other computer systems, are areas where much research is needed.

3. Hypertext

3.1 Hypertext for interface consistency

Work on the Justus system [Wilson 90a, 90b, 91,92] has shown that a simple hypertext interface can be used for most of the functions listed above: document browsing and drafting, form filling and expert systems, linking diverse information from disparate sources. Variation and, consequently, inconsistency in the interfaces for the different legal tasks is greater than it need be. The task in which hypertext has least facility is information retrieval by associative techniques and boolean query, where it loses out to traditional information retrieval techniques. We might gain insight into why this is by comparing the philosophies and functions of the two systems.

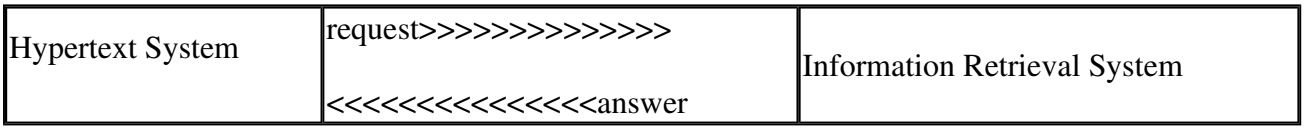
3.2 Hypertext V Traditional Information Retrieval

Traditional information retrieval is about the management of vast quantities of data on secondary storage. Occasionally the data is free text, but more usually it is organised into main data, which is frequently tagged to aid retrieval, and indexes. The chief aim of the system is to provide fast retrieval (access) for many users simultaneously. Multi-user controls are highly developed and facilities to handle concurrent transactions, update the information and define user privilege, are integral. The user formulates a query, which results in data being transferred from secondary storage to main memory. The user is usually aware of the difference between information in primary storage, and the management of information in secondary storage: the system is task oriented.

In hypertext systems the distinction between data in primary storage and data in secondary storage is not clear. The user perceives the information as a set of nodes connected by links. The most usual form of access is by navigation from the current node to a connected node. However, the basic functions of hypertext data management have not yet been defined by international standard and different systems have different concepts and implementations. Consequently, hypertext systems are recognised by a style of user interaction: the system is interface oriented.

3.3 Hybrid systems

Traditional information retrieval systems have attracted massive investment. As engines for storing, updating and retrieving information in a multi-user environment they have achieved an efficiency that none of the current hypertext systems approach. Even if they did, the financial burden of converting the data into a completely different organisational structure would be prohibitive. If we want to extend the advantages of the hypertext interface to the serious user of existing comprehensive databases then, in the short term, we can only hope to achieve this by using hypertext systems as front ends to current information retrieval engines. Thus, the traditional data management system would be responsible for node storing, updating, and retrieval. Entry would either be directly through node name/address or by finding nodes that satisfy qualifying conditions specified by the user. The resulting set of nodes would serve as an initial entry point for conventional hypertext browsing with node retrieval managed by the hypertext system.



The advantages of such a hybrid system are:

i) it can manage large volumes of data with many transactions in a multi-user environment.

ii) navigation and associative access are both available.

3.4 Implementing a hybrid system

Information retrieval systems already exist, so most implementation effort and investment will be required to build the hypertext front-ends. These must support:

1. different task environments in an integrated way: they must be capable of offering a consistent interface for information retrieval, document drafting, and litigation support.

2. interfaces to particular hosts: e.g. LEXIS in UK, BELINDIS in Belgium, JURIS in Germany, etc.
3. various user group sizes. This is of particular importance for activities where group authoring is central to the activity, as in litigation support. Group authoring must be managed by the hypertext system not the data management system; however, because hypertext browsing will greatly increase the number of retrieval requests to the associated data management system, possible degradation in response must be monitored for that system.

Hybrid systems can most easily be implemented if the database is marked-up or tagged so that it lends itself to a variety of different purposes: formatting and printing, content analysis and the collection of statistics, automatic indexing and information retrieval, conversion to hypertext. Consequently, the mark-up must be purely descriptive, i.e. the data must be independent of the purpose for which it is being used. So that the same procedures can be used with confidence on a set of similar documents it must be possible to write a formal description of a class of documents in terms of possible components and the structural relation of the components, one to another. This provides a way of classifying documents through their general structural properties. A mark-up system which supports data independence and document typing is the Standard Generalised Mark-up Language (SGML), [ISO 86]. (Beginners may find [Smith 88] and [Gold farb 90] more digestible.) In SGML such a description is called a *Document Type Definition* (DTD). Parsers exist to ensure that documents tagged in SGML and purporting to belong to a defined document type conform with the relevant DTD, just as compilers for Pascal check that programs written in Pascal obey the grammatical rules of the language.

4. SGML

4.1 Defining document structures in SGML

A function of SGML is to define the structural elements of a text. For example, a simple book structure might be marked up in SGML as shown in Figure 3.

```
<book>
        <title>     </title>
        <author>    </author>
        <imprint>   <publisher>         </publisher>
                    <place>             </place>
                    <date>              </date>
        </imprint>
        <chapter>
                    <chapter.heading>   <number>    </number>
                                        <heading>                   </heading>
                                        <section>   <section.head>              </section.head>
                                                                    <paragraph>
                                                                    <----more   paragraphs-->
                                        </section>
                                        <----more   sections--->
```

```
        </chapter>
        <---more    chapters--->
</book>
```

Figure 3: Document instance of sirnple book in SGML

The *Document Type Defnition* (DTD) for this might be the set of element declarations shown in figure 4.

```
<!DOCTYPE book [
<!ELEMENT book                                  --(title, author, imprint, chapter+)>
<!ELEMENT imprint                               --(publisher+, place*, date?)>
<!ELEMENT chapter                               --(chapter.heading, section+)>
<!ELEMENT chapter.heading                       --(chapter.no, heading)>
<!ELEMENT section                               --(section.head, paragraph+)>
<!ELEMENT paragraph                             --(#PCDATA)>
<!ELEMENT (title/author)                        --(#PCDATA)>
<!ELEMENT (publisher/place/date)                --(#PCDATA)>
<!ELEMENT (chapter.no/heading/section.head)     --(#PCDATA)>
]>
```

Figure 4: Document Type Definition for simple book in figure 3

In the DTD each element declaration has three components:

i) the name in the metalanguage (or the generic identifier) of the elements being declared, e.g., *book, imprint, chapter, chapter.heading, title, author, publisher, place, date, chapter.no, heading, and section.head.*

ji) the minimisation rules, which show whether start and end tags must be present in every occurrence of teh element. In the simple example above, every minimisation rule has the form -- showing that both tags must be present. However, since nothing can follow a paragraph except another paragraph, or the end-tag for a section, it might be argued that the end-marker for <paragraph> is redundant and the declaration might be re-written:

```
<!ELEMENT paragraph -0 (#PCDATA)>
```

Similar arguments apply to section and chapter.

iii) the content model which specifies the content of the element in terms of reserved words or other elements. For example:

```
<!ELEMENT paragraph -- (#PCDATA)>
```

Here, #PCDATA is a reserved word standing for Parsed Character DATA, which means that the element paragraph can contain any sequence of valid characters.

```
<!ELEMENT chapter.heading - - (chapter.no, heading)>
```

This shows that chapter.heading is composed of two elements: *chapter.no* and *heading*. The elements are linked by a comma. A comma used in this way is a group connector. Group connectors

in SGML are:

, components linked by a comma must both appear in the order specified
& linked components must both appear but may appear in any order
| only one of the linked components may appear.
<!ELEMENT imprint -- /cross(publisher+, pjace*, date?)>

The symbols +, * and ? are occurrence indicators. They show that an imprint is composed as follows:

+ at least one occurrence of the element publisher but the element may be repeated any number of times
* the element place is optional but if it does occur it may occur any number of times
? the element date is optional.

4.2 Attributes In SGML

Attributes in SGML distinguish information which is descriptive of an element from information integral to it: attributes specify properties not content. Attributes have names and data-types which must be declared in the DTD. Attributes which might be valuable in the context of hypertext include:

i) *identifier* or ID attribute
Every element in the DTD is potentially a hypertext node so every element should have an ID attribute as a means of attaching a unique identifier.

ii) *type*
Gone are the days when law could be regarded as a largely textual system with an occasional table. The expanding use of the computer in litigation support and for presenting complex evidence to courts means that legal hypertext Systems must be able to handle graphics, images, audio and video.

iii) *owner*
The owner has authority to update the element. If the owner is not the same as the creator a creator attribute may also be required.

iv) *date*
This shows when the element was created. The date may not be a required attribute for all elements, but should always be included for the DOCTYPE element.

v) *permissions/protections*
This helps to ensure information security on the system.

4.3 Non-structural elements in SGML

In the DTD above only structural elements were described. For hypertext, non-structural elements are of equal importance because they provide the non-hierarchical links that are the distinguishing feature of hypertext. Some of the more important of these are described briefly below.

i) Intra-document citations
This is a reference to another element in the same document. The element must have a unique ID attribute. Suppose in Chapter 8 we had a reference:

See Chapter 5

In SGML this might become an in.link element with a IDREF attribute:

See <in.link rid = cH4> Chapter 5 </in.link>

ii) Inter-document citations
IDREF attributes cannot be used across documents because they must be resolvable within the containing document. Therefore we need an ex.link element with attributes that specify what the target is: for example,

<ex.link target.doc = WLR target.element = 1978.1.705>

Target document identifier and target element within document are the minimum attributes. In a distributed hypertext system we might need more, including:

- host
- location of host on network
- target document location, file, or path.

In some circumstances it might also be useful to have:

- target document version number
- target element version number.

iii) Multiple versions
Since some legal text is highly dynamic, itis advisable to include a generic element type version. This could be inserted in any element and could include attributes such as:

- version number
- date of creation
- creator/modifier

The content of the version element might include ver. link type elements to the previous version and to the subsequent version if they exist.

5. Design considerations for a hypertext front-end

5.1 Node design

Genuine legal texts are not authored as hypertext systems: they are free text systems to which hypertext conventions must be matched. System implementors have, therefore, no control over node size or over the style and variety of cross-references (links) which the free-text authors choose to use. This means that a hypertext browser for legal documents will work more naturally if the system is chosen to accommodate the conventions of the text: the text can never be constrained by the limitations of the system. Hence, an SGML tagged document is more naturally mapped to hypertext by mapping every element in the DTD onto a hypertext node. The elements also correspond with the units of information that might be cross-referenced or cited. This has two consequences:

i) In some documents the size of the smallest nodes will be much less than the area available for display, i.e. the normal window size: in Statutes the textual segments are labelled down to sentence level, and occasionally below; hence, most of the basic (atomic) textual nodes will be small. In other documents the largest of the nodes may be too large to fit into normal window size: in law reports, ajudge's opinion might well run to several pages of printed text. This means that the advantage for natural representation will lie with scrolling systems: card based systems will need a mechanism to divide a textual unit, defined as indivisible by the DTD, into card-sized chunks with links to next and

previous.

ii) Since a whole document may also be regarded as a node, systems that can easily handle the aggregation of nodes into larger nodes will be advantageous. All systems should support trees and lists.

5.2 Screen management

If a single interface is required it must be simple: perhaps only two windows; certainly no more than four. System designers have often over exploited technology: multiple overlapping windows or excessive tiling simply confuse the novice or irregular user. Many lawyers fall into this category. If the interface can be tailored then several user groups might be targeted: novices, occasional users, experts, and project managers.

i) Novices or students will need explanation of how the system works and an extremely simple windowing layout. They will primarily be interested in browsing and simple query creation.

ii) Irregular or infrequent users may primarily want to retrieve information in electronic or printed form. Such users may well require less help and explanation than beginners but they will still appreciate a simple uncluttered screen.

iii) Regular users might well generate original documents by combining their own notes with information from databases in the system. They might well be knowledgeable about computers in general and this system in particular and, ideally, might be allowed to specify what style of windowing system they want.

Presentation choices might include:
a) Should windows be cascaded (i.e. multiple overlapping windows) or tiled (no overlapping but rapid reduction in window size)?
b) Should a new window be created:

- for every node access?
- for every different document accessed?
- for every different document type accessed, e.g. separate windows for statutes, for law reports, for precedents etc?
- in a distributed hypertext system, for every different host accessed?

Expert users must, therefore, have access to window management functions - open, close, enlarge, reduce, iconise, de-iconise. In addition to the browsing and query functions available to beginners they might well want a more complex query facility, perhaps with a condensed format and abbreviated commands. They might want to ask for view changes, or the re-composition or reformatting of retrieved information for other purposes. They might want to set up links within their own text, add annotations to nodes belonging to their group, and suggest other editing/linking changes to the system manager.

iv) The manager of a multiple user project in an area like litigation support will need to define management structures for data input locally and set data and system function privileges and protection codes, i.e. decide who can perform what operations on which objects. In this type of project, a hypertext system which supports multiple functional levels is essential.

5.3 Response Time

Rapid response to information requests, either through query or while browsing, is the most important factor in promoting hypertext usability. However, in a hybrid system, navigation is no longer a simple matter of accessing and displaying another node within the hypertext system. Time may also be needed
i) to initiate and execute a program to translate a hypertext signal into a request to the information system
ii) for the retrieval program to find and return the information
iii) to convert the tagged text into hypertext.

Obviously, this will work best if the amount of information retrieved in response to a single request is small. Requests for large documents in their entirety will lead to system degradation. In a distributed system, it may also be necessary to access a remote host. There are two obvious ways to minimise the effect on efficiency:
i) To have a means of backgrounding complex requests while continuing with other work. To this end it would be advisable to treat every request as a new node within the local hypertext system. This would also help to integrate the activities of searching and browsing.
ii) For very large texts, it may be necessary to store skeleton structures in hypertext form, while leaving the content in the main database.

5.4 Query Specification

In a hybrid system, the type of query that can be supported will depend on the interrogation engine of the data management system. Hence, whether the system will handle boolean or non-boolean queries, whether it will support only exact matches or regular expression matching, whether it will rank the nodes retrieved, are all functions of the information retrieval system, not the hypertext front end. All the front-end can do is offer a variety of methods for query construction. Instead of merely allowing the query language of the database host it could offer a choice between query by command language or query by form with pull-down help screens and indexes. Flow diagrams for representing boolean queries may also help naive users: Figure 5, [Schneiderman 9l].

Figure 5: Shnejderman's flow diagram for representing boolean quefies

6. Conclusion

From the brief survey above it can be seen that design of a DTD suitable for diverse tasks is not easy. This paper has concentrated on criteria to ensure that an SGML document can be converted into good hypertext. (Experience has shown that the criteria for good hypertext are more stringent than those for efficient information retrieval, see [Wilson 92].) Hypertext applications are proliferating as stand-alone packages and as front-ends in hybrid systems Thus, it is here that it would seem there is most to gain from consistency and coherence in the databases underlying all systems. In the short term it may be easier to hand-craft a small hypertext or expert system than it is to define DTD's for the documents and then design converters that will workfor all documents of the same type. However, in the long term all advantages lie with the well designed system, with a single, portable database that can be used for diverse purposes.

Among the advantages that accrue are:

i) There is a single database to maintain. This database is independent of the task in which it is used, but, if well designed, can have the flexibility and coherence to serve many diverse procedures: electronic data transfer, high quality printing, information retrieval, the creation of hypertext.
ii) Every document in the database is an instance of a document type with a formal specification. The formal specification can be used as a template for new documents.
iii) Efficient information retrieval can continue through existing search engines.
iv) High quality printout can be available on demand.
v) Compatibility between the nodes and links needed in hypertext and the elements defined in the formal specification can facilitate easy conversion to produce stand-alone systems or browsing front-end interfaces for more user-friendly systems.

Acknowledgments

I would like to thank Dr. G. Raines and Mr. R.C. Saunders for reading and advising on the first draft of this paper.

Bibliography

[BILETA 91] BILETA, *Report of the BILETA Inquiry into The Provision of Information Technology in UK Law Schools*, ("The Jackson Report"), CTI Law Technology Centre, University of Warwick, 1988.

[Bing 80] Bing, J. & Selmer, K.S. *A Decade of Computers and Law*, Universitetsforlaget, Oslo, 1980.

[Bing 86] Bing, J., *Legal Text Retrieval Systems - The Unsatisfactory State of the Art*, in Access to Legal Databases in Europe: Proceedings of the Eighth Symposium on Legal Data Processing in Europe, Strasbourg, 1986.

[Bondi 91] Bondi T., *A Future for Courtroom Technology*, Computers and Law Vol.2 Issue 5, November, 1991.

[Campbell 84] Campbell, C., (ed) *Data Processing and The Law*, Sweet & Maxwell, London, 1984.

[Capper 88] Capper, P. & Susskind, R.E., *Latent Damage Law - The Expert System*, Butterworths, London, 1988.

[Colson 90] Colson, F., & Rule, J., *An Introduction to HiDES*, 5th Bileta Conference, Warwick, 1990, pp.37-41.

[Edwards 90] Edwards, L. & Huntley, J.A.K., *Creating a Civil Jurisdiction Adviser*, 5th Bileta Conference, Warwick, 1990, pp.43-76.

[du Feu 80] du Feu, D. "Selecting welfare benefits by computer" in Computer Science and Law, Niblett B. (ed), Cambridge University Press, 1980.

[Goldfarb 90] Goldfarb, C.F., *The SGML Handbook*, Oxford University Press, Oxford, 1990.

[Hammond 86] Hammond, P., *Representation of DHSS Regulations as a Logic Program,* Computing

Department, Imperial College, London, Report number 82/86.

[ISO 86] International Organization for Standardization, Information Processhg - *Text and Office Information Systems -Standard Generalized Markup Language (SGML)*, ISO 8879: 1986(E).

[Leith 91] Leith, P., Paliwala, A. & Jones, R., *Siting CAL in a Legal Environment: CMLCAL*, 6th Bileta Conference, Durham, 1991, pp.35-43.

[Matthews 91] Matthews, J., *A New Image for Law Firms: Document Imaging in The Legal Profession*, Computers and Law, Vol.2 Issue 4, 1991.

[Mowbray 85] Mowbray, A., *LES: Legal Expert System Generation User Manual, DATALEX Project*, University of New South Wales, 1985.

[Nichols 91] Nichols, S.J., *Law Databases 1991*, Aslib, 1991.

[Painter 90] Painter, D., *Hyperlaw*, 5th Bileta Conference, Warwick, 1990, pp.113-122.

[Paliwala 91a] Paliwala, A., *Creating an academic environment. The development of technology in legal education*, Yearbook of Law, Computers and Technology, Vol.5, 1991.

[Paliwala 91 b] Paliwala, A., *Information Technology in Legal Education: a Resource Book*, CTI Law Technology Centre, University of Warwick, 1991.

[Schneiderman 91] Schneiderman, B., *Designing Advanced User Interfaces*, International Federation for Information Processing WG9.1 Conference on Jobs and Computer Interfaces, Tampere, 1991.

[Sergot 86] Sergot, M.J. *et al* "The British Nationality Act as a Logic Program" 29(5) *Comm ACM,* 1986.

[Smith 88] Smith, J.M. & Stutely, R., *SGML: The User's Guide to ISO 8879*, Ellis Horwood Ltd., Chichester, 1988.

[Susskind 87] Susskind, R.E., *Expert Systems in Law*, Clarendon Press, Oxford, 1987.

[Wilson 90a] Wilson, E., *Cases for Justus: preparing a case database for a hypertext information retrieval system*, Journal of the Association for Literary and Linguistic Computing, Vol.5, No.21 1990, pp.119-128.

[Wilson 90b] Wilson, E., *A Guide to Justus: an overview of a hypertext legal database*, 5th Bileta Conference, Warwick, 1990, pp.163-I73.

[Wilson 91] Wilson, E., *Enabling Draftsmen: a hypertext interface for automated document drafting*, Law Technology Journal, Vol.1, No.1,1991.

[Wilson 92] Wilson, E., *Using hypertext as an interface to the Oxford Advanced Learners' Dictionary,* Computers and Language 2: Towards 1992, to be published in the Office for Humanities Communication Series. 1992.

[Yoder 89] Yoder, E., *Using Hypertext in a Law Firm*, Hypertext 89 Proceedings, Pittsburgh, Pennsylvania, November 1989.

Author
Eve Wilson
Computing Laboratory
University of Kent at Canterbury
Canterbury
Kent CT2 ThJF
England
Tel: +44227764000 ext. 3628
Fax: +44 227 762811
Email: ew@ukc.ac.uk