LEGISLATIVE KNOWLEDGE BASE SYSTEMS FOR PUBLIC ADMINISTRATION - SOME PRACTICAL ISSUES

Peter Johnson, SoftLaw Corporation, Canberra, Australia
David Mead, SoftLaw Corporation, Canberra, Australia

Abstract

This paper looks at three aspects of the use of knowledge base systems which model provisions of legislation:

The first part of the paper suggests that knowledge base systems can be viewed simply as a means of changing the medium in which legislation is presented to administrative decision-makers for their consideration. When viewed as a delivery vehicle for a body of rules, rather than as a reasoning decision-making tool, knowledge base systems which reliably model legislation offer major benefits in the administration of complex legislation.

The second part of the paper looks at some of the practical issues which arise when building a production version of a large-scale knowledge base which models legislation. We suggest a methodology, which we use in constructing and maintaining large-scale systems. The issues discussed in this section have profound implications for the selection of the tools used to build large knowledge bases from legislation: the tool must provide facilities which allow reliable methods of construction, verification and maintenance.

Finally, we suggest that generalist shells are usually structurally inappropriate for the development of large expert system applications in the legal domain. As has happened in other domains, the legal domain must develop domain-specific tools for the creation of large knowledge base systems.

Introduction

This paper is intended as a practical paper, summarising several of our conclusions from building and implementing legal knowledge base systems in Australian government departments.

Systems which we have built are now operational in the Australian Taxation Office, the Department of Social Security and the Department of Veterans' Affairs. Each of these systems is substantial, using at least 500 rules involving several thousand decision points. The largest of the systems, in the Department of Social Security, has over 2500 rules.

These applications have been built using STATUTE, a knowledge base management system which we designed specifically for expert system applications based on legislation or systems of administrative rules.

This paper does not deal with the crucial issue of knowledge representation. STATUTE uses an English language knowledge representation scheme. [Johnson and Mead].

Part 1: The Place of Knowledge Base Systems in the Public Administration of Legislation

The Significance of Poor Administration of Legislation

Problems with widely dispersed public administration of legislation are well-known: inconsistent, arbitrary and otherwise poor decisions. Primary decision-makers in a large government department may at different times lack the training, the experience, the talent or the will to properly apply the legislation which that department administers. [1]

These problems, which are essentially managerial rather than legal, nevertheless lead to a breakdown in the rule of law. There may not be a calculated flouting of the rule of law, such as would raise cries about public corruption. Rather, there is a sad resignation concerning the difficulties of administering legislation in a large bureaucracy.

[1] Other factors would include the complexity and scale of the legislation, and the difficulties with access to, interpretation and use of Case Law. [Taylor and Browne], p.2.

The people who are directly affected by that legislation may well have recourse to elaborate systems of administrative review. However, the truth of the matter, at least in Australia, is that the vast majority of people adversely affected by poor decisions do not seek review of these decisions. The more marginalised and less articulate is the person affected by the decision, the less likely he or she is to have recourse to tribunals, courts or systems of internal review. Needless to say, where a member of the public is incorrectly favoured by a poor decision, he or she is very unlikely to seek review.

In these situations, the rule of law has broken down. Rights are not regulated according to law but according to a mix of caprice, prejudice, ignorance and office lore. The problem is not one which should be viewed as purely managerial. Where the administration of legislation is unreliable, there is a substantial legal problem.

Basic Problems in the Administration of Legislation

There are a range of factors which can lead to the incorrect application of legislation. Problems of malice and prejudice are largely beyond the scope of this paper. We wish to concentrate on problems of interpretation of legislation. These problems are more likely to be endemic in a large bureaucracy than are problems of malice, and, we suggest, can be substantially overcome with the use of knowledge base technology.

We suggest that there are three common interpretative problems which confront a public servant administering complex legislation.

1. The first of these problems is the fragmented nature of legislation. Lawyers are familiar with legislative organisation and structure. The following set of statements discusses only legislative structure (rather than content) yet it should be coherent to a lawyer:

   > The primary provision is section 117, which draws on interlinking definitions from subsection 3(1). One such definition is qualified by subsection 3(10). Section 117 is also qualified by subsection 118(5) and section 122, while sections 126 and 136 contain exceptions to the general application of section 117.

   While such a complex set of interlinking provisions may be mother's milk to a lawyer, it presents problems to a non-lawyer. The truth of the matter, for both lawyer and non-lawyer, is that the legislation cannot be properly applied unless the whole of the legislation is understood. If only a portion of the legislation is understood, a crucial but hidden provision may go unnoticed.

2. The second problem is the internal logical complexity of particular provisions. Long provisions frequently use nested paragraphs and sub-paragraphs linked by a network of words such as "and", "either", "or", "notwithstanding", "unless", "other than" and "subject to".

   It is often difficult to discern the correct logical structure of such a provision at first glance.[2] This is particularly true for non-lawyers, who may be unused to the format and conventions of legislation. A small mistake in this process of structural interpretation, such as treating a disjunction as a conjunction, misinterpreting the order of evaluation of logical expressions or failing to recognise a double negative, can have dire consequences, as is shown in [Allen and Saxon].

3. The third problem of statutory interpretation is that of interpreting particular words or phrases. This is the province of judgement. Because of its reliance on judgement, this area is less susceptible to clear-cut answers than are the previous two problems.

The last problem above is too easily thought of as the sole or major problem in the application of legislation. Questions of the correct interlinking of fragmented legislation and of the internal logic of a provision will often be unambiguous and present no difficulty to an expert. While this may be true for lawyers, our experience has been that those first two types of problem are at least as common in an administrative context as the last problem.

The traditional administrative solution to these three problems of interpretation is to develop policy manuals, which re-organize, paraphrase and expand on the legislation. These manuals are designed to assist the primary decision-makers in administering the legislation. In practice, the manuals frequently become the first and only reference point of the decision-makers.

A range of problems arise from this administrative solution. Manuals are difficult and expensive to continually update in any rapidly changing environment. Over a period, the sheer volume of information makes the manuals unwieldy, and the updating process prone to error. Finally, management discovers that staff rarely actually use the manuals, relying more on their and their colleagues' accumulated wisdom to deal with virtually all situations.[3]

---

[2] This is shown vividly in [Allen and Saxon], pp. 96-102.

[3] These problems associated with manuals have also been recognised in England, see [Taylor and Browne], p. 9.

In our experience, much of the need for manuals derives from the first and second of the interpretative problems outlined above. We have worked with a range of Australian Government Departments. Examination of the policy manuals of each of these Departments generally discloses that the bulk of the interpretative material deals with problems of legislative layout and legislative logic, rather than with the micro-interpretative problem of the meaning of particular phrases.

Finally, we note a trend in the modern administration of legislation in Australia: the phenomenon of "returning to the legislation". In this process, government departments (or politicians) decide that good administration of legislation will only be possible when their staff are equipped to administer the provisions of the legislation directly, rather than through the use of some secondary source of the law.

This process of "returning to the legislation" may involve the rewriting of legislation in a more coherent form.[4] The Australian Department of Social Security has recently completed such a task, while the Australian Taxation Office and the Department of Veterans' Affairs are currently undertaking such "legislative simplification". The process may also involve training the staff in basic principles of statutory interpretation and encouraging confidence in the direct use of legislation.

Encouraging the use of primary legislative material does not overcome the essential difficulties in applying legislation. Legislation of any scale will inevitably contain the seeds of difficulty in interpretation: an elaborate and elegant structure which requires competent navigation; complex provisions or sets of provisions, which contain networks of exceptions, qualifications and extensions; and phrases which require careful consideration and the exercise of judgement.

The Use of Knowledge Base Systems in the Application of Legislation

Knowledge base systems can substantially overcome the first two of the problems identified in the previous section: the problems of legislative layout and the internal logic of provisions. They can also assist in very flexible ways with the administration of discretions and matters of judgement, which legislation will inevitably contain.

Knowledge base systems are often looked at as a decision-making tool. We prefer to look on them more as an administrative tool, which can structure the process by which legislation is considered and decisions made. It is this function of decision support which we have pursued in our development of knowledge base systems for Australian government departments.[4a]

This may appear a modest use of the technology. Certainly, it substantially avoids many of the theoretical problems of modelling legal knowledge. It does this by characterising the knowledge base system as an adjunct of a human decision-maker, rather than as a complete decision-making facility. In this way, the software is absolved of much of the responsibility for the application of law to facts. The provision of a decision support facility is also more palatable to the administrative workforce than is the creation of a decision-making facility.[4b]

We change the medium of the legislation which primary decision-makers are to administer. The present medium is the hard-copy legislation. Working with this medium, the decision-maker is not competent to apply the legislation until he or she has fully comprehended the material - the structure, logic and meaning of the complete legislation.

Knowledge base systems allow legislation to be presented to the decision-maker in a step-by-step format. A system can accurately model all of the structure and logic of many pieces of legislation.[5]

The breakdown of legislation into discrete issues and the presentation of those issues to primary decision-makers in discrete steps highlights the extent to which problems of statutory interpretation for the non-lawyer are structural. In short, much legislation which appears intimidating and complex when presented as a whole is rendered straightforward when presented as discrete questions.

Our experience has been that the presentation of complex legislation in a step-by-step format, via knowledge base modelling of the legislation, has the following consequences:

- the decision-makers have little or no difficulty in dealing with the bulk of the provisions while directly applying the subject legislation;
- the process of consideration of issues is necessarily more thorough; and
- the process of consideration of issues is necessarily more consistent.

Modelling legislation, so that it is presented to the user in a step-by-step format is not always possible. Some types of provisions present great difficulty. In addition, modelling

---

[4] This can significantly reduce the amount of statutory material, and remove the need for much of the interpretative rule base. See [Brown], Appendix III.

[4a] For a fuller discussion of some of the ways in which knowledge based systems can be applied to legal practice, see [Greenleaf], [Greenleaf et al 1988].

[4b] See [Magnusson].

[5] See [Sherman], For futher issues in modelling the logic of legislation, see [Routen].

of legislation does not overcome the problem of open texture issues which are contained in the legislation (such as discretions or matters clearly calling for judgement). Our approach to open texture issues is not discussed in this paper.

The modelling of legislation in knowledge bases does not solve all of the problems in the administration of legislation. However, the practical application of knowledge base technology has much to offer in the legal field, even if used modestly as a decision-structuring tool rather than as a decision-making tool. The immediate usefulness of the modelling process in simplifying legislation is apparent when such a knowledge base is viewed. Moreover, when the aim of the modelling process is simply to change the medium in which the legislation is presented, the process can be demonstrated to be reliable.

Our experience in developing large knowledge bases on legislation has also led us to a range of conclusions concerning the appropriate method for rulebase development and some necessary elements of and design criteria for the shell which is used. These are discussed in the second and third parts of this paper.

Part 2: Practical Issues in the Development of Large Legislative Rulebases

This part of the paper is chiefly concerned with the practicalities of application design. We have built large legal knowledge bases which model legislation and have maintained one of these knowledge bases since late 1988. This work was founded on the prototyping of small and large knowledge bases between 1984 and 1988.

We learned that the development of large knowledge bases for production systems is not simply a matter of scaling up from small experimental knowledge bases. The construction and maintenance of large legislatively-based knowledge bases introduces a range of issues which are not immediately apparent when working with small knowledge bases.

The problems which we discovered in the process of scaling up concerned issues such as the richness of the knowledge representation scheme being used, difficulties in verifying and maintaining large systems and the generation of coherent reports from large systems. These issues are discussed in detail in [Johnson and Mead].

A large legislative rulebase, which is designed to be used in the field by a public body charged with the administration of legislation, must have several characteristics:

- the rulebase must be legally correct;
- the rulebase must be verifiable;
- the rulebase must be maintainable; and
- the rulebase must form part of an application which is practically useful and attractive to the people who have to use it.

The combination of these basic specifications has, for us, led to conclusions concerning the following:

- the appropriate form of knowledge representation for such systems;
- a desirable structure for the knowledge base;
- the relationship between the knowledge base and the application development environment;
- the features of the application in which such a knowledge base must reside; and
- the method by which the rulebase which models the legislation is constructed.

The issue of knowledge representation is discussed in [Johnson and Mead]. The other issues listed above are discussed in more detail in the following sections of this paper.

Rulebase Method

We have developed and refined a method for constructing rulebases from legislation which satisfy the criteria listed above. Our experience has been that departure from this method leads to substantial difficulties. The method is nothing more than good legal method, transferred to software construction.

The method which we use is based upon the following principles:

1. as far as possible, verbatim modelling of the precise terms of the legislation;

2. rejection of modelling any "overview" of the effect of a piece of legislation and of any shortcuts in the complete and verbatim modelling of the structure and content of the subject legislation.

3. explicit modelling of each level of the structure of the legislation;

4. a strict separation of rules which explicitly model the terms of the legislation (legislative rules) from those which model some interpretation of the effect of the legislation (interpretative rules) and again from those which model some wider body of general knowledge (common-sense rules);

5. a rigorous separation of the legislative rulebase from the machinery which drives the end application.

These principles of method allow the production of large legislative rulebases which are reliable, maintainable and verifiable. However, these methods are not able to be followed if the tool for construction of the rulebase does not allow them. These principles make substantial demands of that tool: the form of knowledge representation, the relationship between the rulebase and the procedural language which drives the application and the structure of the question search and inferencing mechanisms.

We developed the STATUTE Knowledge Base Management System (KBMS) to meet these and other demands. Other more general purpose shells may be capable of meeting the requirements of this method by adding additional tools to the development environment.

Each of the listed principles is explained, with examples, below.

1. Verbatim Modelling of Legislation

Verbatim modelling of legislation means including in the primary rulebase all of the subject legislation and only the subject legislation.

A rulebase developer may decide to leave out small portions of the legislative material for a number of reasons. The most common problem among people whom we have trained is the tendency to jump to modelling interpretative material without first creating the explicit legislative foundation for that material in the rulebase.

A second problem is the desire to paraphrase a piece of legislation, because the legislative material is seen as needlessly turgid or obscure.

A third problem is the tendency to leave out "machinery" - legislative phrases which internally connect one provision to another. This is discussed under the next point "No Overviews, No Shortcuts".

A large rulebase which models legislation but which does not model it verbatim introduces problems when the time comes for additional construction, verification and maintenance. These problems grow significantly as the rulebase grows and as it increasingly departs from the terms of the legislation. [6]

[6] For a discussion of the negative effect of having other than verbatim modelling, based on experience with Scandinavian systems, see [Schartum].

The problems of verification and construction may be manageable when modelling a small piece of legislation. In that case, the knowledge being modelled is really the knowledge which a human expert has of the effect of the legislation. It can be said, in a pure theoretical sense, that any construction of a rulebase based on legislation involves modelling human knowledge of that material. However, in the practical world of constructing a workable application, the verification and maintenance of that rulebase are demonstrably more reliable when each component in the rulebase is immediately and obviously referable to the fixed subject legislation, rather than to a less reliable human "overview" of the legislation.

The issue of verbatim modelling immediately raises the question of knowledge representation. It is impossible to comply with this principle of verbatim modelling if the knowledge representation scheme which is being used will not allow this.

When we commenced our work on rulebase legislative systems, we used a conventional symbolic form of knowledge representation:

Example:
```
elig_invalid_pension if
        yes age_requirements
and     yes residence requirements
and     yes health_requirements
and     yes special_requirements.
```

After having built a large knowledge base using such a scheme of knowledge representation, our conclusion was that the scheme was dangerously crude, and that the rules would have to be written in complete English sentences. This conclusion was forced upon us by considerations such as the richness of the knowledge representation and problems with verification, maintenance and reporting. These issues are fully discussed in [Johnson and Mead].

The use of the English language as the scheme of knowledge representation allows verbatim modelling of the terms of legislation. This immediately enhances the capacity of the rulebase developer or any independent expert to verify the correctness of the rulebase visually.

Once the developer has the capacity to use full English sentences to represent facts and premises, there is a further issue of the rigour with which the legislation is modelled verbatim. This is discussed in the next point.

2. No Overviews, No Shortcuts

Overviews of the effect of legislation, and shortcuts in the modelling of legislation are inviting and usually reliable in small systems.

In large systems, failure to model the legislation verbatim has invariably led us into problems when it became necessary to add to a rulebase or to amend a rulebase. Our experience has been that only rigorous and verbatim modelling of the legislation allows ready verification and modification.

The rigour which we believe is prudent includes:

- verbatim modelling of provisions such as "except in cases where section 32 applies" (rather than the shortcut of immediately modelling section 32);

- verbatim modelling of the structure of provisions;

  Example: The claimant satisfies section 54
  if
  The claimant satisfies paragraph 54(a)
  and
  The claimant satisfies paragraph 54(b)
  and
  The claimant satisfies paragraph 54(c);

- avoidance of overviews which restructure the content of legislation to fit the accepted administrative view of the effect of that legislation, replacing verbatim law with approximating lore;[7]

- tedious incorporation of each step in a protracted legislative chain; these are particularly prevalent where series of nested definitions are used;

  Example: The claimant satisfies section 36
  IF The claimant was in an operational area
  The claimant was in an operational area
  IF The claimant served in an
  area described in a scheduled item
  The claimant served in an area described in a scheduled item
  IF The claimant served in an area described in schedule 1.
  The claimant served in an area described in schedule 1 IF . . .

In this last example of nested definitions, the incorporation of each step in the legislative path appears pointless and extremely tedious to the rulebase developer. However, such rigour is justified when the developer later finds an exception to one of the steps in the chain, or when the rulebase is submitted for independent evaluation, or when the legislature amends one of the nested definitions.

Once again, rigorous compliance with principle throughout the construction of a large rulebase not only renders it more transparent, reliable and verifiable,[8] it saves major headaches later.

3. Explicit Modelling of the Structure of Legislation

Verbatim modelling of legislation is the modelling of the exact terms of the legislation. The extent to which and the manner in which the implicit structure of the legislation is modelled is a separate issue.

Much of the difficulty of interpreting legislation derives from the complexity of its structure. Elegant and well-drafted legislation reduces this difficulty by identifying discrete logical components through accepted organisational devices: paragraphs and sub-paragraphs, explicit references to discrete qualifying provisions, definitions etc.

These devices provide convenient handles, which are constantly used in statutory interpretation. The following mental processes are familiar to any lawyer, though no substance is discussed:

> Paragraph 117(1)(c) is an alternative to 117(1)(d).
> Subsection (1) is to apply in all cases, except those outlined in subsection (2).
> Paragraph 117(1)(e) is now subject to the exception introduced by section 117A.

A rulebase which simply models substance and which fails to explicitly model structural components will introduce several difficulties:

- verbosity, where elegantly nested legislative alternatives are each expanded, modelling every permutation of the set of alternatives;
- difficulty in verifying the rulebase, because permutations must be exhaustively assessed to check whether the structural logic has been given effect;
- difficulty in maintaining the rulebase, where one or several legislative units (such as sub-paragraphs) have been amended and where these units do not each occupy a discrete place in the rulebase; and
- reduced capacity to convey the source of a rulebase outcome, because of the failure to report the specific legislative provisions satisfied.

---

[7] See [Schartum] for problems occurring if this happens even to a slight degree.

[8] That transparency and ease of verification are essential as discussed in [Bing], which also shows the value in having direct access to the actual words of the legislation.

By modelling structural components of the legislation, the rulebase developer incorporates the organisational elegance of the draftsman into the knowledge base, which assists with construction, verification and maintenance. It often has the further effect of assisting the end-user to understand the manner in which the subject legislation has or has not been satisfied.

Example: The claimant is eligible for age pension if
The claimant satisfies paragraph 25(1)(a)
and The claimant satisfies paragraph 25(1)(b)
and Sub-section 25(2) does not apply to the claimant

The rule in the example above does not model the content of the relevant provision, but only structure. However, this ensures that the organisation of the rulebase will mirror that of the legislation. Amendments to the legislation - which are always expressed in terms of these structural components - are able to be incorporated easily into the knowledge base.

4. Separation of Rule Types

Good statutory interpretation separates the function of considering the structure of legislation from the function of considering the meaning of particular words (except where those words relate to the structure), and from the function of considering judicial pronouncements on the meaning of those words. [Rissland and Skalak]. Good judicial interpretation of legislation rigorously ties argument to the terms of the subject legislation.

The structure which a KBMS allows for rule creation will substantially determine the extent to which different functions, and different bodies of knowledge are kept discrete or jumbled together. Once again, we suggest that good organisation is essential if large rulebase systems are to be verifiable and maintainable.

Our method is to develop a complete rulebase which models the legislation, and verify the correctness of that verbatim model, before we attempt to deal with the effect of any judicial material on the meaning and effect of particular provisions.

In order to allow this, the STATUTE KBMS recognises two major types of rules: legislative rules and interpretative rules.[9] This recognition not only allows the developer to neatly separate and examine the legislative modelling from the interpretative material, it also allows the end user to switch on or switch off the interpretative rules.[9a] In this way, the same rulebases can be used in a variety of different ways by users with different levels of knowledge.

An interpretative rule has, as its goal, a particular line of the legislative rulebase. In this way, the connection between the interpretative material and the subject legislation is clear and accurate. The process of maintaining and verifying the interpretative rulebase is made more reliable because of this explicit reference to a particular portion of the legislation.

In addition, this method encourages good practice by the rulebase developer. The different portions of a reasoned judicial argument on legislation must be separated and correctly attributed to the appropriate legislative provisions.

The STATUTE KBMS also allows a third type of rule - the common sense rule. These rules merely carry out background inferences which are based on accepted general knowledge.[10]

Large rulebases require very good organisation if they are to be verifiable and well-maintained. The separation of different functional types of rules assists in the organisation of a large rulebase. We suggest that the three-fold separation which we use is appropriate to the legislative domain and the very least that is required for a large system.

5. Separation of the Rulebase from the Application Machinery

Different types of expert system shells offer the application developer different means of control over the performance of the knowledge base and of the application as a whole.

Crude shells require the rulebase and the procedural instructions which drive that rulebase to reside in one structure. In this case, rulebase components and procedural instructions are mixed throughout a single application program.

Some shells allow limited external control of the rulebase by an application language and external control of the performance of the application as a whole. Many of these have only limited capacity to separate the rulebase structure from the machinery for investigation of that rulebase. So,

---

9 For another system which incorporates both these sort of rules, especially where the interpretative rules are case based, see [Rissland and Skalak].

9a For the system to be acceptable to the end user, both are required, and required to be separated in this way. See for example [Magnusson].

10 Gardner has also recognised common sense rules as important, and incorporated them into her doctoral work. [Gardner 1984], [Gardner 1985].

for example, the order in which lines in the rulebase are investigated depends strictly upon the order in which they occur in the rulebase.

These types of systems, which require the rulebase investigation to follow the rulebase structure, have substantial drawbacks for large-scale systems. They invite a structuring of the rulebase which does not follow the organisation of the subject legislation, but which is organised according to the desired question path.

We believe that it is desirable to use a knowledge base environment which allows the form of the rulebase to be independent from the form of the investigation of that rulebase. The ideal is to be able to structure a rulebase completely on the basis of ease of verification, maintenance and further construction, while having the capacity to investigate the contents of that rulebase according to any desired path.

This requires certain links and facilities to be built into the expert system shell. The question search and inferencing mechanisms must be designed to allow for this level of control of the knowledge base. In addition, the procedural language which is used to build the final application must have appropriate facilities to assist in that control of the knowledge base.

The STATUTE KBMS allows issues of rulebase construction to be completely independent from issues of rulebase investigation. The rulebase is built with an eye to legal integrity, verification and maintenance. The rulebase should mirror the legislation in its terms, its structure, its order and organisation. An additional and separate range of facilities in the question search and procedural components of the KBMS allow the application developer to nominate the ways in which this rulebase is to be investigated - entry points, order of investigation, order in which different legislative alternatives are investigated, tangential investigations to be carried out under certain conditions, side-tracks, short-cuts and stopping points.

Any application developer building a large system is going to need and demand the sort of procedural control outlined above. Where that procedural control is embedded in the rulebase itself, the legal integrity of a large rulebase will be questionable, while maintenance and verification will be difficult. Where that procedural control is spread between an external facility and some ordering of rulebase components, there are serious dangers for the maintenance and verification of the system.

These problems become more substantial as the application grows, and particularly as the application needs to be amended. Limited capacity to separate control of the application from construction of the rulebase leads to a higher level of compromise in the method of rulebase construction.

Part 3: Integration of a Legal Knowledge Base in a Practical Administrative Application

A rounded administrative application could seldom contain a knowledge base in isolation.

If the application is to be useful and acceptable to users and management in a government department, a legal knowledge base application would typically contain the following types of facilities:

- dynamic linking to powerful text retrieval;
- the capacity to generate coherent reports, in a variety of forms acceptable to users and clients;[11]
- the capacity to build flexibility into the application, to allow for different skill levels of end-users;
- the capacity to link to third-party software, such as word processors, databases and spread-sheets.

Integration capabilities of the primary expert system shell with text generators, screen generators, text retrieval facilities, word processors, databases and similar software are crucial. The more dynamic and flexible is this linking capability, the more imaginative is the administrative solution which you can develop.

These issues of integration capability, user interface, linking to text retrieval and linking to third party software are often looked at as cosmetic add-ons. In practice, the extent to which these facilities can be incorporated into an application is often determined by the structure of the expert system shell used. The fundamental design of the knowledge representation scheme, rule creation scheme, inferencing mechanism and relationship between the knowledge base and the application language all have major effect on the capacity to build a practical, maintainable application.

In our experience, these "cosmetic issues" significantly contributed to our complete abandonment of one prototype of the expert system shell, including the knowledge representation scheme, the rulebase design and inferencing structure. These "cosmetic" criteria were fundamental considerations in the final design of the core expert system components of the STATUTE KBMS.

---

[11] For the importance of this in the content of legal research, and computer applications to Law in general, see introduction to [Sprowl].

A STATUTE-based application will typically contain the following components, within a unique user-created application environment:

- a rulebase which directly models the subject legislation, known as the legislative rulebase;
- a rulebase which overlays the legislative rulebase, and which models appropriate secondary material, such as judicial interpretation of particular legislative provisions (the interpretative rulebase);
- a rulebase which co-exists with the legislative and interpretative rules, and which drives background common sense inferences (common sense rules) and knowledge-base instructions to the application language to perform some special tangential function (daemons);
- annotations which link appropriate provisions in the rulebase to portions of textual research materials (legislation, policy manuals, cases, administrative instructions etc) for retrieval during a knowledge base investigation;
- reports which model output, combining rulebase-generated explanation of facts and conclusions, text from secondary sources and free-form text input by the user during the course of the consultation;
- substantial files written in the application language, driving the application;
- access to advanced text retrieval facilities such as hypertext and free-text retrieval; and
- links to the corporate database.

We believe that the legal domain requires specialist shells and methodology which are suited to the domain and the target users. These specialist shells must be developed with an eye to integration of practical tools for administrative or legal solutions, rather than have pure knowledge base creation tools developed in isolation from application creation tools.

References

[Allen and Saxon]
Layman Allen and Charles Saxon, "Some Problems in Designing Expert Systems to Aid Legal Reasoning", proceedings First International Conference on Artificial Intelligence and Law, Boston, May 1987 ACM Press, 94.

[Bench-Capon et al]
Bench-Capon, Robinson Routen and Sergot "Logic Programming for Large Scale Applications in Law: A Formalisation of Supplementing Benefit Legislation." In Proceeding of the First International Conference on Artificial Intelligence and the Law, Boston, May 1987, 190.

[Bing]
Jon Bing, "Three Generations of Computerized Systems for Public Administration and some Implications for Legal Decision Making", (1990) Vol. 3 No. 2 Ratio Juris, 219.

[Brown]
Brown, Shibboleths of Law: Reification, Plain-English and Popular Legal Symbolism. Legal Research Foundation Enc., Publication No. 27 (1987).

[de Bessonet]
Gary de Bessonet, "An Automated Intelligent System Based on a Model of a Legal System." (1984) 10, Rutglers Computer and Technology Law Journal.

[Gardner, 1984]
Anne Gardner, "An Artificial Intelligence Approach to Legal Reasoning, Bradford Books/MIT Press 1984.

[Gardner, 1985]
Anne Gardner, "Overview of an Artificial Intelligence Approach to Legal Reasoning". "In Computing Power and Legal Reasoning" (Walter Ed.), West Publishing Co., St. Paul, Minnesota 1985, 247

[Greenleaf]
Graham Greenleaf, "Legal Expert Systems - Robot Lawyers?", Paper presented at the Australian Legal Convention, Darling Harbour, Sydney, August 1989.

[Greenleaf, et al, 1988]
Graham Greenleaf, Andrew Mowbray and Alan Tyree, "Communications aspects of Legal Expert Systems - Incorporating them in Shells for Lawyers". National Conference on Law, Computers and Artificial Intelligence, University of Exeter, United Kingdom 17-18 November 1988.

[Greenleaf, et al, 1987]
Graham Greenleaf, Andrew Mowbray and Alan Tyree, "Legal Expert Systems: Words, Words, Words". (1987) 3 Yearbook of Law Computers and Technology, 119.

[Johnson and Mead]
Peter Johnson and David Mead, "Natural Language - An Appropriate Knowledge Representation Scheme for Legislative Expert Systems". Unpublished paper, Canberra 1991.

[Kuopus]
Jonna Kuopus, Hallinnon Lainalaisuus ja Automatisoitu verohallinto, "The Rule of Law and Computerised Administration of Taxation, Finnish Lawyers", Publishing Company, Jyvaskyla 1988, with an English Summary.

[Magnusson]
Cecilia Magnusson, "Introducing Knowledge Based Systems in the Swedish Social Insurance Organisation". (1988) Volume II No. 2 Information Age 103.

[Rissland and Skalak]
Edwina Rissland and David Skalak, "Interpreting Statutory Predicates". In Proceedings of the Second International Conference on Artificial Intelligence and the Law, Vancouver, June 1989, 46.

[Routen]
Tom Routen, "Heirarchically Organised Formalisations". In Proceeding of the Second International Conference on Artificial Intelligence and the Law, Vancouver, June 1989, 242.

[Schartum]
Dag Wiese Schartum, "Representation of Legal Rules in Computer Programs - some Implications on Legal Protection and Service to Citizens:. Norwegian Research Centre for Computers and Law.

[Sherman]
David Sherman, "A Prolog Model of the Income Tax Act of Canada". In Proceedings of the First International Conference on Artificial Intelligence and the Law, Boston, May 1987, 127.

[Sprowl]
James Sprowl, "A Manual for Computer Assisted Legal Research", American Bar Foundation, Chicago 1976.

[Taylor and Browne]
Andrew Taylor and Julie Browne, "Supporting Local Office Adjudication: The Plurality of Requirements, the Multiplicity of Options and the Problems of Organisations". Presented at BILETA third Annual Conference, Warwick, 1988.