



5th BILETA Conference British and Irish Legal Technology Association

EXILE - Expert Information In Legal Evidence

P.T.J. Cattell, J.Y. Machtynger and I.A. Wilson

Abstract: Expert systems form a relatively new though somewhat ill-defined class of computer system. While many have been implemented in other than rule-based languages, it is still somewhat challenging to use a language like Prolog to implement an inference machine with a good user interface. EXILE is a project aimed at exploring the issues involved in building a system which is able to give 'advice' on a decision-making issue and to supply an explanation of the decision-making process at any stage during the process of reaching a conclusion or after the conclusion itself is reached. EXILE gives legal advice on whether a 'document' is admissible as evidence in a court of law. The project is part of a continuing research program at the Queensland University of Technology and is a co-operative venture between the Faculties of Information Technology and Law.

1. Introduction to Law of Evidence

The Oxford dictionary defines "the law" as a system made up of rules established in a community and the order produced by that system. Thus, the courts administer laws which control the conduct of society by permitting or prohibiting various types of action by individuals.

Most law administered by the courts is "substantive" law i.e. legal rules which govern various fact situations and which lay down who is "right" and who is "wrong". This is the type of law with which most people are familiar. For example, the criminal law says that it is wrong to steal someone else's property, the law of contract says that a person who breaches his or her obligations under a legally enforceable agreement may have to pay damages to compensate the "innocent" party and the law of negligence says that anyone who breaches their legal duty not to injure those around them may have to pay compensation to their victims. In all of these types of cases it is necessary, before the rules of substantive law can be applied, to determine exactly the facts and circumstances surrounding the case. There is therefore another body of law administered by the courts which is sometimes described as "adjectival" law i.e. it consists of rules which determine not which facts may be proved to a court (this is determined by the substantive law) but how those facts may be proved and how the case is to be conducted from beginning to end.

The vast majority of cases which fall to be decided by the courts involve disputes about the facts. Each side often wishes to present a different version of what is supposed to have occurred. Thus, a large proportion of a lawyer's time is occupied not by matters of substantive law (which are often very clear) but by matters of fact, which may be in dispute either because memories of what happened or what was intended may differ with the passage of time or because someone sets out to deliberately conceal the truth.

The law of evidence is concerned with those rules and principles which affect how facts may be

proved to a court and, more importantly, how facts may not be proved i.e. the law of evidence lays down a number of exclusionary rules which limit the type of material which may be presented in support of a case or against the case of an opponent. Facts are usually proved to a court through witnesses giving sworn testimony i.e. persons who have seen or heard or otherwise experienced the happening of an event relate this to the court on pain of punishment for telling deliberate falsehoods. Another means of proving facts, however, is to place before the court documents which cast some light on what actually happened. The law of evidence has many rules relating to the testimony of witnesses and many others relating to the production of documents.

2. Hearsay in Law of Evidence

One of the most basic rules of evidence is concerned with what is called "hearsay". Hearsay is simply what someone else was heard to say (or seen to write) about a particular subject, and is in general prohibited by the law of evidence. Thus, anything said or written by a person can only be related to the court by that person, not by someone else, if the purpose of relating it is to prove that it is true. The reason for this rule is that a witness gives evidence on oath and is subject to cross-examination to check the veracity of their testimony. Neither of these safeguards would be available to check the trust-worthiness of the maker of a statement or the author of a document if that person were not giving testimony but simply having what they said or wrote repeated by someone else.

The law of evidence is primarily common law, however a number of acts of parliament affect it by modifying or removing some of its aspects. In Queensland the relevant statute is the Evidence Act 1977-1985 although very similar acts can be found in all States and Territories of Australia. The Queensland Evidence Act modifies the common law rule against hearsay described above by creating a number of exceptions to that rule as it applies to documents over and above the exceptions (not relevant here) which already exist at common law. The two most important sections in the Evidence Act and its interstate equivalents which operate to create these exceptions are section 92 and section 93, but some six or so other sections and many more sub-sections of the Act are directly related. It is not proposed here to discuss any of these sections or sub-sections in detail since they are extremely complex and their meaning has been the subject of a very large number of cases in court. It was however the very complexity and, at times, ambiguity of these sections and sub-sections which gave birth to the idea of creating an expert system to assist the practicing lawyer in understanding and applying them to the facts of any given case at hand.

Using traditional methods, a lawyer unfamiliar with these sections and sub-sections would require to expend a considerable number of hours in research before being "expert" in all aspects of their operation. Using an expert system, that lawyer has had the hard research work already done by others who specialise in the field and is able to be supplied with a ready answer as to whether the sections apply to the particular document he or she seeks to place before the court. Since most lawyers operate in an extremely busy and demanding professional environment it was thought that an expert system on a topic such as documentary hearsay which is not only outside the well-known fields of substantive law but in a field of adjectival law outside the mainstream of that discipline would be of considerable benefit. The remainder of this article seeks to canvas not the substance of the law concerned, for this is the subject of an article in itself by one of the authors (Wilson, 1985),¹ but rather the way in which the expert system known as EXILE was constructed to achieve an academically sound, practically useful and somewhat unique legal tool.

3. THE EXPERT SYSTEM APPROACH

There has been much publicity in recent years concerning expert systems. Much of this has been quite ill-informed. One of the more common misconceptions is that any application written in a rule-based language is, per se, an expert system. While some classical expert systems (such as XCON)

have been implemented using rule-based languages (OPS5 in this case), the vast majority of classical expert systems have not been so constructed.

Expert Systems should be primarily characterised in terms that are functional rather than constructional. The term 'expert' refers to someone who through extensive training or through the development of skills after much experience is able to operate in a problem-solving domain with a level of competence much superior to that of the non-expert. An expert system is a computer-based system that is able to emulate or assist an expert in his problem-solving activity.

Another characteristic of an expert is that he often finds it quite difficult to explain precisely how he goes about his task. However, it is usually expected that he attempt to do so, if only to train other potential experts in the same field. As the expert goes through his problem-solving procedure, an observer or learner should be able at any stage to ask why a certain action was taken or to ask for other information relevant to the problem-solving process. When he has reached his conclusion (diagnosis, finding, ...), an observer should also be able to ask how the conclusion was reached. EXILE exhibits both of these characteristics. There have been other attempts to build systems which behave in a similar manner (Walker, 1983).²

An expert system should also be amenable to easy incremental improvement as new knowledge about the problem becomes available. It is partly for this reason that rule-based languages such as Prolog and OPS83 are popular, since they allow the encoding of new information as simply new rules added to the knowledge base without major reconstruction of the rule-interpretation part of the system. New knowledge is able to be made available to EXILE by adding new rules, or by deleting or modifying existing rules. A further advantage of this rule-based approach is that knowledge is able to be handled in relatively small disjoint chunks, with rule alteration having minimal effect on the rest of the knowledge base and usually no effect at all on the inferencing part of EXILE.

A related advantage of rule-based knowledge bases is that they allow more exact emulation of the way in which most experts express their knowledge. In many areas, an expert is able to make statements such as the following.

'In this situation, I do this.'

or

'If these conditions hold, then this follows.'

In other rather more rare applications, the essence of the problem is actually given as a set of rule-like statements! This is the case with EXILE. The starting point is the "Evidence Act 1977-85" of the State of Queensland³ which is couched as a set of clauses and sections in the usual legal terminology. With some effort, the clauses may be directly translated into rules in the knowledge base. In some trivial cases, this translation simply generates one rule in the knowledge base for a clause in the Act. More commonly, clauses generate one or more rules in a complex fashion with some rules appearing in the knowledge base as representations of more complex legal relationships.

It thus seemed to us that a worthwhile project would be to evaluate the suitability of a rule-based language for this type of application. The choice of language was influenced by a number of factors. One of our prime aims was to build an intelligent explanation system. It is well known that this is considerably more realisable using a goal-driven, backward-chaining approach to rule manipulation. Production system languages are inherently data-driven and forward-chaining, though it is possible to use such languages to build goal-directed systems by manipulating goals as working memory elements. If a functional language were used, much of the behavioural characteristics of a Prolog interpreter would need to have been constructed to give the required rule-interpretation sequence.

It was also expected that the structure of the Act and the inter-relationships between the sections and clauses of the Act would display a relatively rigorous logical structure - with terms and situations defined in one clause being used as pre-conditions in other sections and clauses. It was also noted that the prime result of any particular execution of EXILE should be intrinsically a 'yes' or 'no' decision as to whether the piece of evidence under examination was or was not admissible in a court of law.

For these and other reasons not entered into here, Prolog was chosen as the implementation language. A Sun 3/50 workstation was available to carry out the software development phase using a version of Prolog known as Iforolog.⁴

The Evidence Act consists of over one hundred sections. It was decided to restrict the pilot implementation to sections 92 and 93 which covers aspects of admissibility of documents as evidence in a court of law. The term 'document' covers much more than might appear at first to a non-lawyer and is defined precisely in the Act.

Our expert in interpretation of the Act and co-author of this article (Ian Wilson) is a barrister and lecturer in the Faculty of Law at the Queensland Institute of Technology. He also supplied extra information where necessary to supplement the Act - such as case references which defined legal precedent and directions on where in the Act to find legal definitions of terms used in EXILE. The implementation of code was carried out mainly by a student from the Faculty of Information Technology working in close collaboration with a senior staff member from the same faculty.

4. Types of Knowledge in EXILE

EXILE explicitly and implicitly manipulates a variety of types of knowledge. Firstly, in developing EXILE there was the need to distinguish clearly the difference between algorithmic and definitional concepts in various parts of the Act. Some rules, for example, specify some or all aspects of what a term means rather than how to classify an item as an instance of a term. To illustrate, there are terms used in interrogating a user which have 'legal' meanings that may not be obvious to the layman. These need to be used during dialogue with EXILE but are not defined by rules in the knowledge base. Using the dictionary option, the 'definitions' of these terms may be retrieved. In many cases, the entry in the dictionary is merely a list of typical items that are instances of the term i.e. a non-exhaustive list of examples. In other cases, a definition based on a case precedence is given i.e. a common law definition. In either of these approaches, there will often be the need for the user to use his judgement to decide whether a certain item is an instance of a term or not. It is largely for this reason that EXILE is seen to be an assistant to a lawyer rather than a replacement. Similar distinctions were encountered in formulating rules for the knowledge base.

Secondly, knowledge needs to be encapsulated which allows for concepts developed in one part of the knowledge base to be modified in other parts of the knowledge base. In Section 92, clause (1) specifies conditions for a document to be admissible as evidence in proceedings other than criminal proceedings. Clause (2) modifies the interpretation of clause (1) and clause (3) modifies the application of clause (2)! This interdependence between rules somehow also needed to find representation in the knowledge base.

Thirdly, a great deal of knowledge about the state of dialogue with the user needs to be maintained in order to emulate the sort of explanations which would be given by a human expert in the interpretation of the Act. A major part of the remainder of this article is concerned with outlining how this is achieved.

5. The Development of the Knowledge Base

The knowledge base was derived from the sections and clauses which make up the Act. Subsection 92 (2) is outlined below.

2) The condition in subsection (1) that the maker of the statement or the person who supplied the information, as the case may be, be called as a witness need not be satisfied where -

- a) he is dead, or unfit by reason of his bodily or mental condition to attend as a witness;
- b) he is out of the state and it is not reasonably practicable to secure his attendance;
- c) he cannot with reasonable diligence be found or identified;
- d) it cannot reasonably be supposed (having regard to the time which has elapsed since he made the statement, or supplied the information, and to all circumstances) that he would have any recollection of the matters dealt with by the statement he made or in the information he supplied;
- e) no party to the proceeding who would have the right to cross-examine him requires his being called as a witness; or
-) at any stage of the proceeding it appears to the court that having regard to all circumstances of the case, undue delay or expense would be caused by calling him as a witness.

The word 'or' appearing at the end of paragraph (e) of the sub-section above is read in law as making each paragraph an alternative to all the others i.e. each paragraph is disjunctive.

5.1 User Interface

One of the initial tasks for the EXILE authors involved deciding on a way of reducing the legal verbiage - at least internally in the program - to a manageable level. A Prolog procedure 'express' was written to enable any internal shortened form to be expanded to the actual wording used in the Act. For example, using the actual wording in 92(2) (f) above, the supplier of a statement need not be called to give evidence if

"undue delay or expense would be caused by calling him (the supplier of the statement) as a witness".

This was shortened to the internal form:

supplier is_ not_worth_it".

An 'express' predicate to carry out the conversion from the internal to the external form (eg for displaying text on the screen) could be

express(supplier is_ not_worthit):-

```
write('undue delay or expense would be caused'),
write('by calling the supplier of the statement as a witness.').
```

In truth, express in EXILE is more complex than the above as it also carries out variable left-margin indentation to allow easy interpretation of the responses to queries about how EXILE is reasoning.

As a further level of complexity, for every concept in EXILE, there are three external forms used during user dialogue. There is the interrogative form when the program is enquiring whether something is true; the affirmative form when a statement is made as part of the explanation for how a conclusion was found to be true; and the negative form allowing statements to be made as part of the explanation for how a conclusion was found to be false. At the time of writing, EXILE contains

explicitly three external representations for each internal representation. As part of on-going work, a transformational grammar is to be developed to carry out conversions from the second of these to the other two.

5.2 Rule Generation

Rules were then extracted from the subsections and clauses using the shortened forms of expressions as rule antecedents. This is not a trivial process as, for example, one subsection may give rise to a number of rules as is the case with subsection 92 (2). An example of one of the rules (Rule 8) extracted from this subsection follows below.

It deals with the conditions under which the supplier of a statement need not be called as a witness.

```
rule8    if    supplier is_dead
          or    supplier is_ out_of_ state
          or    supplier is_ hard_to_ find
          or    supplier has forgotten
          or    supplier is_ not_ needed
          or    supplier is_ not_ worth it
then
          supplier need not_be_called_1
```

This is the exact form of the rule in the knowledge base. It can be seen that even a casual reader can interpret the general meaning of the rule quite successfully. More importantly, a lawyer with little or no experience in programming, but with a little experience with EXILE should be able to understand the rules easily. Indeed, he should also be able to formulate new rules, delete rules, or modify rules to improve the knowledge base or to expand the scope of EXILE should there be amendments to the statute or court decisions affecting it.

A long-term aim of the EXILE project is to allow a lawyer to insert new rules using wording very close to the original act and to have EXILE carry out its own generation of internal representations. This will enable an examination of the difficulty of extracting distinct concepts from the original wording and an analysis of the decision-making process regarding the best method of representing such concepts.

The method in EXILE of implementing the supporting structure so rules similar to the one above can be directly interpreted as legal Prolog syntax is interesting but is not addressed in this paper.

Rules may also be implemented where they have a set of antecedents connected by and's rather than or's. An example of this is rule 14 which was extracted from clause 93 (1) (a).

93. Admissability of documentary evidence to facts in issue in criminal proceeding.
[cf. Eng. 1965 S. 1; WA. 5. 79E.]

1) In any criminal proceeding where direct oral evidence of a fact would be admissible, any statement contained in a document and tending to establish that fact shall, subject to this Part, be admissible as evidence of that fact if -

- a) the document is or forms part of a record relating to any trade or business and made in the course of that trade or business from information supplied (whether directly or indirectly) by persons who had, or may have reasonably be supposed to have had, personal knowledge of the matters dealt with in the information they supplied -

```

rule14  id    dealing with s92
        and   statement isin document
        and   direct_oral_evidence is_ admissable
        and   document forms part_of_record
        and   record made in_course_of_undertaking
        and   record made from_personal_knowledge
        and   supplier will be_called
then
        prove8 is_ok

```

In this rule, prove8 refers to the eighth way in the act by which a statement may be shown to be admissable. This rule also is in a format which is easily able to be understood or modified by a lawyer with no training or experience in programming.

6. The 'How' and 'Why' of EXILE

One important feature of most expert systems which differentiates them from other classes of computer program is their ability to justify conclusions. Any conclusion reached by EXILE can be explained by asking 'how' it was reached. 'Why' EXILE is asking a particular question of the user prior to reaching a conclusion can also be explained. This attempts to emulate the way an expert might react to questioning while he is going about his task. A non-expert who is supplying information to the expert might well ask questions such as the following.

```

"Why are you asking me that?",
"Why is that relevant here?",

```

or

```

"How did you arrive at that decision? I want to see your logic."

```

EXILE, upon being questioned in this way, is able to react sensibly to such 'why' and 'how' questions.

6.1 Manipulation of Facts

Prior to outlining examples of EXILE's explanation process, it is necessary to indicate how EXILE manipulates facts. For a more detailed discussion of the different types of knowledge that are both explicit and implicit in the building of expert systems and the relationships between them, the reader is directed to a paper by Tom Addis (Addis,1987)⁵. The simplest class of facts in EXILE are facts that may be discovered by simply asking the user whether they are true or not. These are implemented as 'askable' predicates in Prolog as outlined by Bratko (Bratko,1986)⁶.

EXILE has the ability to infer facts from other facts that have been previously inferred or told to it, and then to use these new facts to help establish other facts. For example, the following rule (after expansion) appears in the knowledge base:

```

rule1    if      the matter is a proceeding
        then
            we are dealing with section 92

```

So, if in some section of our knowledge base, we have another rule with the following partial statement:

.
 .
 and we are dealing with section 92
 and
 .

then, provided that rule 1 can be successfully asserted, EXILE is able to use this deduced fact to help satisfy the requirements of the second rule.

It is by combining supplied facts and deduced facts in a network of rule interdependence that EXILE handles the complexity of the Act in a logical, structured manner. This network (or inference net) may be transformed conceptually into an and-or tree if all rules (and their corresponding sub-trees) which have their deduced facts used as more than one rule antecedent are replicated in the network.

6.2 'Why' a Question is Being Asked

Owing to the goal-directed behaviour of EXILE, it needs to start with an initial goal. This goal is normally the top-level goal

"This document is admissible as evidence."

though it is possible from a menu to choose as an initial goal one of a number of goals which are sub-goals of the top level goal. From this initial goal, EXILE is able to generate all sub-goals that need to be proved in order to reach a conclusion about the truth or otherwise of the initial goal. If a sub-goal in the and-or tree is reached for which there is no rule in the knowledge base, the list of askable predicates is consulted. If the sub-goal is able to be tested by asking the user to supply a 'yes' or 'no' answer to a question, then this is done.

Clearly, at the stage when a question is asked, the user needs to respond. EXILE allows the user to respond with 'yes', 'no', 'why' or a number of other options. If the response is 'why', the inference chain, from the leaf of the tree that generated the question up to the initial goal, is generated. This is then expanded using the wording of the Act and formatted to produce an explanation of why the question is being asked of the user.

To demonstrate the internal workings of these functions, the following scenario is presented. Suppose that the user wants to know whether something is admissible as evidence. During the query session, he is permitted elaboration of words using the dictionary and justification of EXILE's questions with the use of 'why'. Here is a sample screen output illustrating how 'why' responses are handled.

Is the matter a [proceeding] other than a [criminal proceeding]?
 (y/n/w/h/ag/e): why

So I can investigate by using rule 1, that we are dealing with section 92,
 so I can investigate by using rule 4, that the statement is admissible according to S92(1),
 so I can investigate by using rule 17, that the statement is admissible as evidence,

which was your original question.

Is the matter a [proceeding] other than a [criminal proceeding]?
(y/nlw/hlag/e):

Note here, that when the user asks 'why' a question is being asked, EXILE shows all the sub-goals that are currently under examination in attempting to satisfy the initial goal, from the point in the inference net where the question is being posed. In the case above, the initial goal was

'The statement is admissable as evidence.'

and the first primitive sub-goal reached in satisfying that goal gave rise to the first question asked. If, instead of replying 'why' in response to the first question above, the user had answered 'no', an alternative dialogue could have been as shown below.

Is the matter a [proceeding] other than a [criminal proceeding]?
(yln/w/h/ag/e) :no

Is the matter a [criminal proceeding]?
(y/n/w/h/ag/e): yes

Is the statement you wish to tender in written form?
(yln!w/h/ag/e): why

So I can investigate by using rule3, that the statement in question is contained in the form of a document,
so I can investigate by using rule16, that the statement is admissable according to s93,
so I can investigate by using rule17, that the statement is admissable as evidence,

which was your original question.

Is the [statement] you wish to tender in written form?
(y/n/w/aq/e):

Note that the 'why' proof has now changed, firstly since the immediate question has changed and secondly, since a different course of questioning is followed if the matter is a criminal proceeding.

Is the [statement] you wish to tender in written form?
(ylnlw/ag/e): yes

Does the subject matter of the [document] contain [admissable opinion] or statement of [fact]?
(ylnlw/h/ag/e) :yes

Does the document form all or part of a [record]?
(y/n/w/h/ag/e) yes

Does the [record] relate to, and was it made in the course of trade or [business]?
(y/n/w/h/ag/e) yes

Was the [record] made from information, supplied by person or persons who had or may reasonably be supposed to have had personal knowledge of the matters dealt with in the information they supplied?
(y/n/w/h/ag/e) yes

Is the supplier dead or unfit by reason of bodily or mental condition to attend as a witness?

(yln/w/h/aqle) yes

I have found an answer to your problem.

The statement is admissable as evidence.

6.3 'How' a Conclusion Was Reached

When EXILE has reached its conclusion, the user may interrogate it as to 'how' it reached its conclusion. As EXILE searches for a way to prove (or disprove) its initial goal, every goal (whether as a fact it was told to be true or as a fact it deduced to be true) is appended to the head of a list with a record kept of how the fact was found to be true. Sometimes, a line of inference may reach a point from which further deductions are impossible. It then backtracks, removing entries from the head of the list until an alternative line of inference is found. Successful deductions along this new line are then added to the list. For a detailed discussion of the implementation of this type of explanation system, see (Bratko,1986)⁶.

This building and pruning of the list continues until EXILE reaches its final conclusion. The user is then able to ask how EXILE reached its conclusion. If the user does request the explanation, a complete chain of sub-goals proved and the method of their proof is able to be outlined by simply extracting them from the list.

By looking at a well known syllogism, the logic of EXILE can be demonstrated.

If	Socrates is a man
and	men are mortal
then	Socrates is mortal

Assuming that EXILE was operating with such a rule in its knowledge base, it would firstly attempt to find rules to prove that

Socrates is a man
and men are mortal.

On finding no such rules, it would ask the user to tell it whether each of these was true. If the user supplied the information that they were true, it would reach the conclusion

Socrates is mortal.

Upon the query 'how' being asked by the user, EXILE's proof of its conclusion would be

Socrates is mortal
was ->derived from:
Socrates is a man
was ->told to me

and
Socrates is mortal
was ->told to me

In the same way, EXILE using its backward-chaining behaviour generates all possible primitives (ie questions for which answers must be told, rather than derived). An example explanation from EXILE is given below.

The statement is admissable as evidence

was->derived by rule17 from:

the statement is admissable according to s93 was ->derived by rule16 from:

we are dealing with section 93

was ->derived by rule2 from:

the matter is a criminal proceeding

was ->told to me

and

the subject matter of the document contains opinion considered admissable, or fact

was ->told to me and

the document forms part or all of a record was ->told to me

and

the record relates to, or was made in the course of business was ->told to me

and

the record was made from personal knowledge

was ->told to me

and

the supplier of the statement need not be called according to s93(1)

was ->derived by rule9

from:the supplier of the statement is dead or until by reason of bodily or mental condition to attend as a witness

was ->told to me

Thus the entire proof has been displayed specifying how all the deduced facts were obtained from the information that was told to EXILE. The importance of such an outline of how a conclusion was reached cannot be over-emphasised. It allows any user to examine EXILE's internal behaviour, to learn from this, or to correct it if need be, and is an important pre-requisite of successful knowledge engineering.

7. EXILE Dictionary

EXILE has a dictionary of relevant legal terms. There are two places from which definitions of words may be called. The first is from the main menu. This allows one to use EXILE simply as a reference on current interpretations of various legal terms. The second is from within an EXILE query. In this way, elaboration of highlighted words used by EXILE is possible. Highlighted words are those enclosed within square brackets in this paper.

Is the matter a [proceeding] other than a [criminal proceeding)?
(y/n/w/h/ag/e) explain

(h/your-word/<return>/)> proceeding

Proceeding means any civil, criminal, or other proceeding or inquiry, reference or examination in which by law or by consent of parties, evidence is or may be given, and includes an arbitration.

In the above example, 'proceeding' and 'criminal proceeding' are highlighted using square brackets. When using the EXILE dictionary, the user may refer to any word resident within it, not just those words that may be highlighted at the time. A help menu exists showing which words are currently stored. There is no real limit other than storage restrictions to the number of words EXILE can explain and their interpretations can be updated frequently if necessary.

8. Conclusions

The choice of Prolog as a development language for EXILE gave some immediate advantages but, unfortunately, some other significant disadvantages. The major advantage lay in the implementation of the knowledge base. It was easy to generate a knowledge base which was elegant and uncluttered. Some clauses in the Act map onto exactly one rule in the knowledge base. Each rule is expressed both in an easy to read form and in strictly legal Prolog syntax making errors easy to detect and correct. Implementation of other major sections of the Act in an expanded knowledge base is now quite feasible. It is hard to imagine any language tool which would better support this part of the project.

However, the implementation of the inference engine and user interface is complex and was very time-consuming to fully debug. Often, a task which would have been simple in an algorithmic language held up development while an artifact was built in Prolog to achieve the desired result. Some of these delays may have been 'once-only' overheads as the authors gained familiarity with some aspects of Prolog. Certainly, now that the explanation process is functional, that part of the system needs only relatively minor additions to improve its functionality.

The pilot version of EXILE, as it currently stands, is a useful tool for lawyers to sharpen their expertise if they are unfamiliar with these sections of the Act and their application. Of particular use are the case histories displayed where necessary to define judgments on issues which are not adequately covered by the wording of the Act. This aspect of EXILE alone has the potential to save many hours of searching by practicing lawyers. As more of the Act is implemented, EXILE holds promise of becoming a useful commercial product. In time, it is intended to further expand EXILE to other areas of legal application.

The inference engine of EXILE has recently been combined with a new knowledge base. This application implements rules published by Telecom Australia relating to charges for data communication. In this case, advice is given as to whether data traffic conveyed partly via a third party network is chargeable by Telecom or not. Another application is also being tested at QIT relating to the design of industrial switchboards subject to the rules of the electricity supply authority and the requirements of the relevant local government.

References

1. **WILSON, I.**, *"Documentary Hearsay: The Scope of the Queensland Evidence Act"*, QIT Law Journal, p.111, Volume 1, 1985
2. **WALKER, A.**, *'An Inference Engine Which Explains Both Yes and No Answers'*, in Proceedings of the 8th IJCAL, Los Altos, CA: William Hoffman, Inc., 1983, pp 526-528

3. Evidence Act 1977-1985 of the State of Queensland

4. [Ifprolog details not available]

5. **ADDIS, T.**, "*Meaning; the Frontier of Informatics*", presented at INFORMATICS 9, held at King's College, Cambridge, 26th to 27th March 1987

6. **BRATKO, I.**, "*Prolog Programming for Artificial Intelligence*", Addison-Wesley (1986), ISBN 0-201-142244