

# AUSTRALIAN COMPUTER JOURNAL

Volume 23, Number 1, February 1991

## SPECIAL ISSUE ON INFORMATION TECHNOLOGY AND THE LAW II

- 1 Guest Editorial — IT Applications to the Law
- 2 The use of object oriented principles to develop intelligent legal reasoning systems  
*G VOSSOS, T DILLON, J ZELEZNIKOW, G TAYLOR*
- 11 Legal expert systems: The inadequacy of a rule-based approach  
*J POPPLE*
- 17 The applicability of a knowledge-based system to legal education  
*SD GREGOR, HM RIGNEY, JD SMITH*

## INFORMATION SYSTEMS

- 22 Australian involvement in electronic data interchange — 1989  
*PMC SWATMAN, JE EVERETT*

## SPECIAL FEATURES

- 34 Guide to authors
- 35 Book reviews



Published for Australian Computer Society Incorporated  
Registered by Australia Post, Publication No. NBG 1124



# Legal expert systems: The inadequacy of a rule-based approach

J. Popple

Department of Computer Science  
Australian National University

---

*The two different categories of legal AI system are described, and legal analysis systems are chosen as objects of study. So-called judgment machines are discussed, but it is decided that research in legal AI systems would be best carried-out in the area of legal expert systems. A model of legal reasoning is adopted, and two different methods of legal knowledge representation are examined: rule-based systems and case-based systems.*

*It is argued that a rule-based approach to legal expert systems is inadequate given the requirements of lawyers and the nature of legal reasoning about cases. A new, eclectic approach is proposed, incorporating both rule-based and case-based knowledge representation. It is claimed that such an approach can form the basis of an effective and useful legal expert system.*

*Key Words and Phrases: case-based systems, expert systems, law, legal reasoning, rule-based systems.*

*CR Categories: I.2.1, I.2.3, I.2.4, I.2.5, I.2.6, J.1.*

---

Copyright © 1991, Australian Computer Society Inc. General permission to republish, but not for profit, all or part of this material is granted, provided that the ACJ's copyright notice is given and that reference is made to the publication, to its date of issue, and to the fact that reprinting privileges were granted by permission of the Australian Computer Society Inc.

Manuscript received January 1990; revised October 1990.

Earlier versions of this paper appear in the proceedings of the Thirteenth Australian Computer Science Conference (Popple, 1990a) and the International Conference on Computing and Information (Popple, 1990b).

## 1 INTRODUCTION

The range of computer applications in the law is wide. It extends from general applications, of use to lawyers, to applications designed specifically for the law. This paper is concerned only with those systems that make use of artificial intelligence (AI) techniques to solve legal problems.

Legal AI systems can usefully be divided into two categories: legal retrieval systems and legal analysis systems.

*Legal retrieval systems* allow lawyers to search through databases, containing details of statutes and decided cases, for information. AI techniques may be employed to simplify this task (e.g. by searching for keywords which have not been input by the user but are deducted to be equivalent to, or sufficiently related to, the input keywords).

*Legal analysis systems* take a set of facts and determine the ramifications of those facts in a given area of law.

(McCarthy (1980a) identifies a third category of legal AI systems: integrated legal systems. He cites as an example computerized title registration systems which make decisions about people's rights and obligations. It is hard to see why such a system could not be usefully classified as a legal analysis system, albeit with some of the features of a legal retrieval system.)

Mehl (1959) claims that there is no fundamental difference between these two categories (legal retrieval systems and legal analysis systems)—that the difference is one of degree only. However, Shannon and Golshani (1988) point out that the difference between systems based on a "conceptual model of legal analysis" and text-retrieval systems is that the latter do not "understand" any area of the law.

This paper will be concerned with legal analysis systems.

## 2 LEGAL ANALYSIS SYSTEMS

Legal analysis systems can be divided into two categories:

- *judgment machines*: systems that make a judge-like pronouncement (e.g. "X is guilty of offence Y for the following reasons . . ."); and
- *legal expert systems*: systems that provide advice similar in form to that which a solicitor might provide (e.g. "The facts in this case are similar to those in *P v D* where the defendant was found guilty, but the instant case can be distinguished from *P v D* as follows . . .").

### 2.1. Judgment Machines

The idea of a *judgment machine* was raised over

thirty years ago by Mehl (1959). Although such a machine would perform the functions of a judge, it was said that a role for humans remained because:

... the solution to a legal problem may depend upon extra-rational factors, involving the whole of human experience ... (p. 758)

Almost twenty years later, D'Amato (1977) suggested that a judgment machine could *replace* a human judge. His proposed machine would take the relevant facts as input and produce a number in the range -1.0 to 1.0 (where a positive number indicates a victory for the plaintiff). Given the multiplicity of factors, he claimed, a result of zero would be extremely unlikely. Somewhat begrudgingly, he allowed for some vestige of human control. An appeal court could review all of the machine's determinations in a certain numerical range (*e.g.* -0.05 to 0.05) within which the cases would be so close that a re-examination might be required. The review court's subsequent decision would then be incorporated into the system.

The idea of human judges being replaced by machines has been vehemently criticized. According to Weizenbaum (1976):

The very asking of the question, "What does a judge ... know that we cannot tell a computer?" is a monstrous obscenity. That it has to be put into print at all, even for the purpose of exposing its morbidity, is a sign of the madness of our times.

Computers can make judicial decisions ... They can flip coins in much more sophisticated ways than can the most patient human being. The point is that they *ought* not to be given such tasks. They may even be able to arrive at "correct" decisions in some cases—but always and necessarily on bases no human being should be willing to accept.

... What emerges as the most elementary insight is that, since we do not now have any ways of making computers wise, we ought not now to give computers tasks that demand wisdom. (pp. 226–227)

Moles (1987) agrees:

The computer scientists, encouraged by the modern positivists, fail to recognize ... that law, positive morality and ethics are inseparably connected parts of a vast organic whole. Judgments are involved at every stage of the legal process and machines cannot make judgments. In stating that legal rules can be applied without further judgment; that they apply in an all or nothing fashion; that legal decision making follows the form of the syllogism or that it is a pattern-matching routine, the modern positivists, joined now by the computer scientists take us along a dangerous road. (p. 271)

But D'Amato sees advantages in replacing human judges by machines:

Would we lose a judge's "judgment," and how important would such a loss be to our legal system? Surely computers do not make "judgments" the way humans do, and

so we would lose the "human" aspect of legal judgments. But what specifically do we lose when we lose the humanness of judgments? Is human judgment just a euphemism for arbitrariness, discretion, or bias? (p. 1281)

Proponents of the idea of automated judges claim that such systems would reduce the cost of the legal system, find inconsistencies in the law, and provide a level of certainty in the law which does not exist at present (as a result of freely-available, automated, judicial advisory opinions).

In 1977, D'Amato claimed that his proposal for a computerized judge was a modest one. Yet, the current state of AI technology is such that no judgment machine has been implemented which can pass judgment in any substantial area of law. The ethical question of whether judges *ought* to be replaced by machines remains a hypothetical one. The author's sympathies lie with Weizenbaum and Moles. For that reason, and because AI technology has not yet produced judgment machines—machines which may prove impossible to build—this paper will, henceforth, be concerned only with legal expert systems.

## 2.2 Legal Expert Systems

For the purposes of this paper, a *legal expert system* (LES) will be defined as a system that provides answers to legal questions which are in a form that one would expect from a lawyer.

This definition excludes AI systems which might merely be used as tools by a lawyer in coming to legal conclusions or preparing legal argument (*e.g.* a sophisticated legal retrieval system). This is not to say that a lawyer should not be able to use an LES, merely that the output from an LES should be usable without further legal analysis. This output should be in such a form that it can be the basis of a lawyer's legal argument in court.

The term *expert system* has become ambiguous. It has variously been defined as a system that deals with "knowledge" gathered from an expert, a system that can perform at the level expected of an expert, or a system that can be used by an expert. Applying any, or all, of these definitions, an LES—as defined above—is an expert system.

LESS are not judgment machines; they will not usurp judicial power (although their imprudent use could lead to the relinquishment of some judicial discretion). The development of sophisticated LESS will not remove the need for lawyers, but it may change the nature of some legal work.

## 3 LEGAL REASONING

All LESS must be capable of legal reasoning, or (at least) of simulating legal reasoning. So all LESS must be based upon a model of legal reasoning.

Any attempt to define legal reasoning soon raises the question of whether that definition is descriptive or normative. McCormick (1978) points out that a theory of legal reasoning can be both descriptive and normative—that it can be normative in its own right, and describe norms that actually operate within the legal system.

The model of legal reasoning adopted in §3.2 is both descriptive and normative. Before describing that model, it is necessary to explain the concept of *open texture*.

### 3.1 Open Texture

The term *open texture* was first used in jurisprudence by Hart (1961). According to Shannon and Golshani (1988):

Roughly speaking, a concept is open-textured if it defies complete definition. (p. 312)

Here we refer to the inherent indeterminacy of the meaning of words that are used to describe the predicates of statutes. Sometimes other factors, such as vagueness, may also be considered as part of the open texture issue. (p. 312, n. 10)

The problem of open texture in legislation is one which arises in the development of any LES. This problem can be demonstrated by reference to McCarty's TAXMAN project (1980a), and to criticism of McCarty's approach to open texture.

The TAXMAN project was concerned with the area of corporate tax law. The basic "facts" of a corporate case were captured in a relatively straightforward representation (*e.g.* a corporation issued securities). Below this level was an expanded representation of the meaning of various entities (*e.g.* a security interest) in terms of their component rights and obligations. Above this level—presumably above *both* levels, although this is not made clear—was the "law" (statutory rules which classify transactions as taxable or non-taxable *etc.*). Legal analysis, according to McCarty, is a simple matter of applying the "law" to the "facts".

McCarty accepts that some legal concepts are open-textured, which makes them very difficult to represent. Moles (1987) argues that *all* legal concepts are open-textured—that even those concepts which McCarty claims are easily represented are impossibly complex (in the sense that they are beyond the capability of machines). Moles complains that:

McCarty appears not to appreciate that 'corporations', 'securities', 'property', 'dividends' and so on are not subsumed 'beneath the law', but are each the products of complex legal analysis. The question of whether certain transactions are taxable or not is intimately tied into that legal analysis. (p. 270)

The sad thing is that he has not shown the slightest awareness of the nature of the legal expertise. (p. 269)

Moles is opposed to the very idea of an LES, but his attack on McCarty's approach is important because it emphasizes the problem that open texture poses for LES design. If all legal concepts are seen as being open-textured, as Moles would have it, then building an LES would be impossibly complex. An LES builder must choose an arbitrary level of abstraction above which a concept may be open-textured, and below which a concept must be considered to be fully defined.

The model of legal reasoning adopted in §3.2 allows that some concepts in a statute may be open-textured, but assumes that these concepts are amenable to full definition by reference to case law.

### 3.2 A Model of Legal Reasoning

For the purposes of this paper, the following model of the process of reasoning with statutes and case law will be adopted.

A lawyer examines the facts of the case in question (the *instant case*), and determines which statutes (if any) apply. These statutes are applied to the facts of the instant case. The meaning of a concept in a statute may be open-textured, and may determine the result of the application of that statute to the instant case.

A lawyer argues about the meaning of an open-textured concept by reference to the facts of the instant case and those of previously-decided cases. The results of some cases are desirable in that they ascribe a meaning to an open-textured concept which (when the statute is applied) leads to a desired result in the instant case. No two cases can be completely identical, given the plethora of facts associated with any given case. Some of these differences may be insignificant, and much of a lawyer's reasoning by analogy concerns the *legal* significance of these differences.

A lawyer argues with cases in the following fashion:

- If the result of a previously-decided case is desirable, they argue that there are no legally significant differences between the previous case and the instant case, so the previous case should be followed.
- If the result of a previously-decided case is undesirable, they argue that there is some legally significant difference between the previous case and the instant case upon which the previous case should be distinguished.

## 4 KNOWLEDGE REPRESENTATION

The method of representing legal knowledge in an LES depends upon whether the LES is concerned with *statute law* or *case law*.

#### 4.1 Statute Law

A number of projects have focussed on representing the provisions of a statute as a set of rules. When these rules are applied to the facts of a case (*i.e.* by instantiating previously free variables) an inference engine can produce an answer which represents the effect of the statute on the given facts. For example, the *British Nationality Act* has been encoded as a PROLOG program (Sergot, Sadri, Kowalski, Kriwaczek, Hammond and Cory, 1986).

Shannon and Golshani (1988) claim that extracting the rules from a statute in an *ad hoc* fashion is unsatisfactory because:

- subsequent designers/users cannot trace the evolution of a set of rules from the words of the statute;
- the rules cannot be mechanically checked for correctness; and
- this *ad hoc* approach may lead to rule formulations which do not work well together.

Although it is not possible to check such rules for correctness, it must be remembered that lawyers are similarly unable to check their own interpretation of a statute for correctness. It is up to the knowledge engineer to check that the rules are an accurate representation of the statute. Shannon and Golshani suggest that it is unlikely that a mechanical method can be developed for transforming statutory language into formal rules, but methods have been developed which (if followed rigorously) reduce the likelihood of error. The use of such methods would also reduce the scope for dissimilar rule formulations. The first problem (being unable to trace the evolution of the statute to the rules) can be obviated by the sensible use of comments.

#### 4.2 Case Law

No area of law is covered exclusively by statute. Even a new statute, which has not specifically been the subject of any case, is interpreted in the light of previously-decided cases. Case law (or the *common law*) is fundamental to the Australian legal system, which relies heavily upon the *doctrine of precedent*. This doctrine states that each decided case is not merely an example that later judges may choose to follow, or to ignore: that case, itself, becomes part of the law. This means that any useful LES must take account of the law embodied in previously-decided cases.

The problem of representing case law is different from, and more complex than, that of representing statutory provisions. If a statutory provision is open-textured, the courts give meaning to that provision. When faced with this open-textured concept, the LES builder has two options:

- to pose the question to the user ("What is the meaning of this open-textured concept?"), and to accept the user's answer as an accurate statement of the law; or
- to incorporate, in the knowledge base, expert knowledge as to the meaning that the relevant cases ascribe to the open-textured concept.

The first option is satisfactory only if the LES is being used by a legal expert who is (presumably) in a position to answer the question. This approach would surely reduce the LES's usefulness. Shannon and Golshani (1988) opt for a combination of both approaches:

This model allows some room for reasoning with the facts but relies on the user for input when no clear inference is found. We add depth to our model by filling in the basic rules and definitions of the statute with additional factual examples from decided cases. (p. 311)

This simplistic solution to the problem of open texture has severe limitations, as discussed in §4.2.1.

Tyree, Greenleaf and Mowbray (1988), with their FINDER system, take a completely different approach to the problems posed by the common law. The area of law which they chose to model is based entirely on cases. They claim that the number of decided cases in any given area of law is usually so small that inductive tree generation algorithms cannot be used. Further, they suggest that it is inappropriate to model case law using a rule-based system:

It is not that it is theoretically impossible to write such rules, but that it is not the natural way in which lawyers reason with cases. (p.232)

#### 4.2.1 Rule-based Systems and Case Law

In fact, it is *not* possible to formulate production rules which will adequately represent case law, because such a rule-based system would be of little use to a lawyer. It is not just that rule-based reasoning "is not the natural way in which lawyers reason with cases." Such a system may be capable of producing an answer (possibly with an attached estimate of its probability). But a lawyer is not interested in a definitive answer—even if it is strongly suggested by a long line of legal authority—because it doesn't assist the formulation of their legal argument.

As discussed in §3.2, a lawyer reasons with cases by arguing that there are no legally significant differences between the instant case and a previously-decided case whose result is desirable, and/or that there are legally significant differences between the instant case and a previously-decided case whose result is not desirable.

No amount of reason extraction from an inductive

rule-base will provide the information that a lawyer needs to argue in this fashion. Hence, attempting to reduce the results of previously-decided cases to rules which can be simply added to a statutory rule base is an inappropriate approach to the problem of open-textured concepts. As Tyree *et al.* (1988) state, such an approach does not reflect the way in which lawyers reason about cases. But, more importantly, it makes for an inadequate LES.

#### 4.2.2 The FINDER System

The FINDER system of Tyree *et al.* (1988) takes the following approach to cases. Expert knowledge is used to determine the most important cases in a given (fairly small) area of law, and the attributes which are of legal importance to the outcome of those cases. These attributes are given weights—not by a legal expert, but by examining the extent to which each attribute differs across the cases. Using these weighted attributes it is possible to measure statistical *nearness* (similarity) between the cases.

When the facts of the instant case (*i.e.* those attributes which are of legal importance) are entered, the nearest previously-decided case (the *nearest neighbour*) is ascertained. If the attributes of the nearest neighbour are the same as those of the instant case then the advice is clear. When the attributes of the cases differ, FINDER gives details of the nearest neighbour, and lists the differences. The system also finds the nearest case which reached the opposite conclusion to that of the nearest neighbour (the *nearest other*). That case, and the differences between it and the instant case, are explained. To reduce the chance of giving bad advice, several statistical techniques are employed to ensure that the nearest case is not greatly different from the instant case.

### 5 FURTHER RESEARCH

Two distinct forms of knowledge representation in LESs have been identified. Rules are appropriate for representing statutory law. They can also be used to represent case law, but this approach is inadequate. Alternatively, a set of attributes can be identified for each of the relevant cases. By comparing these attributes with those of the instant case, a statement can be made about the common law as it relates to the instant case.

None of the systems discussed in this paper (and, to the author's knowledge, no previously developed system) has incorporated *both* of these methods of legal knowledge representation: a rule-based system combined with a case-based system.

SHYSTER (Popple, 1990c) is a prototype of an LES which combines a rule-based system with a case-

based system (similar to the case-based FINDER system). When the rule-based system encounters an open-textured concept, the case-based system is employed to produce a legal argument as to the meaning of that concept.

This eclectic approach has a number of benefits:

- It has the advantage of approximating the approach which a lawyer would take when given a legal problem. The rules (derived from a statute) are applied until the meaning of some (open-textured) concept is required. Faced with this problem, a lawyer would turn to the common law in order to further clarify the meaning of the statute. So, too, does SHYSTER: the lawyer's two-stage approach is clearly modeled.
- It goes some of the way towards responding to the complaints of those who believe that an expert system can never adequately simulate legal reasoning (see Moles' comments in §2.1 and §3.1). By taking the search for the meaning of statutes to the common law, this approach avoids some of the problems inherent in a purely rule-based system.
- It is (it would seem) a novel approach to the problem of knowledge representation in LESs. However, the two disparate methods upon which it relies have been separately, experimentally demonstrated.
- It is not subject to the restrictions on the problem domain which bind previous systems to areas of law which are predominantly statute-based, or case-based, but not both.

### 6 CONCLUSION

This paper has discussed previous developments in LES design and has shown how a purely rule-based approach is inappropriate if an LES is to be of use to a lawyer. A better approach (combining rule-based methods with case-based methods) has been outlined and it is suggested that LESs which incorporate this approach will prove to be fruitful objects of research.

### ACKNOWLEDGEMENTS

The author would like to thank Roger Clarke, Peter Drahos, Bob Moles, Malcolm Newey, Robin Stanton and Neil Tennant for their comments and suggestions.

This research is supported by an Australian National University PhD Scholarship (funded by the Centre for Information Science Research) and by a Sigma Data Postgraduate Research Award.

### REFERENCES

- D'AMATO, A. (1977): Can/Should Computers Replace Judges? *Georgia Law Review*, 11(5), pp. 1277–1301. ISSN 0016–8300.  
 HART, H. L. A. (1961): *The Concept of Law*. Clarendon Law Series, Oxford University Press (Clarendon), London.



- McCARTY, L. T. (1980a): Some Requirements for a Computer-based Legal Consultant. Technical Report LRP-TR-8, Laboratory for Computer Science Research, Rutgers University, New Brunswick, New Jersey.  
Appears with minor abbreviations, as (McCarty, 1980b).
- McCARTY, L. T. (1980b): Some Requirements for a Computer-based Legal Consultant. *Proceedings of the First Annual National Conference on Artificial Intelligence (AAAI-80)* Stanford University, pp. 298–300.
- MacCORMICK, N. (1978): *Legal Reasoning and Legal Theory*, Clarendon Law Series. Oxford University Press (Clarendon), Oxford. ISBN 0–19–876080–9.
- MEHL, L. (1959): Automation in the Legal World: From the Machine Processing of Legal Information to the “Law Machine”. *Mechanisation of Thought Processes*, vol. II, pp. 755–787. Her Majesty’s Stationery Office, London.  
Proceedings of a Symposium held at the National Physical Laboratory, 24–27 November 1958.
- MOLES, R. N. (1987): *Definition and Rule in Legal Theory: A Reassessment of H. L. A. Hart and the Positivist Tradition*. Basil Blackwell, Oxford. ISBN 0–631–15342–X.
- POPPLER, J. (1990a): Legal Expert Systems: The Inadequacy of a Rule-based Approach. *Proceedings of the Thirteenth Australian Computer Science Conference (ACSC-13)*, Monash University, Melbourne, 7–9 February, pp. 303–313.  
*Australian Computer Science Communications*, vol. 12, no. 1. ISSN 0157–3055.
- POPPLER, J. (1990b): Legal Expert Systems: The Inadequacy of a Rule-based Approach. In Akl, S. G., Fiala, F., and Koczkodaj, W. W. (editors), *Advances in Computing and Information: Proceedings of the International Conference on Computing and Information (ICCI-90)*, Niagara Falls, Canada, 23–26 May, pp. 348–351. Canadian Scholars’ Press, Toronto. ISBN 0–921627–70–X.
- POPPLER, J. (1990c): SHYSTER: A Hybrid Legal Expert System. Technical Report TR-CS-90-06, Department of Computer Science, Australian National University.
- SERGOT, M. J., SADRI, F., KOWALSKI, R. A., KRIWACZEK, F., HAMMOND, P. and CORY, H.T. (1986): The British Nationality Act as a Logic Program. *Communications of the ACM*, 29(5), pp. 370–386. ISSN 0001–0782.
- SHANNON, D. T. and GOLSHANI, F. (1988): On the Automation of Legal Reasoning. *Jurimetrics Journal*, 28(3), pp. 305–315. ISSN 0022–6793.
- TYREE, A. L., GREENLEAF, G. and MOWBRAY, A. (1988): Legal Reasoning: the Problem of Precedent. In Gero, J. S. and Stanton, R. (editors), *Artificial Intelligence Developments and Applications*, ch. 16, pp. 231–247. Elsevier Science (North-Holland), Amsterdam.  
Edited selection of papers to the Australian Joint Artificial Intelligence Conference, Sydney, Australia, 2–4 November 1987. ISBN 0–444–70465–5.
- WEIZENBAUM, J. (1976): *Computer Power and Human Reason: From Judgment to Calculation*. W. H. Freeman, San Francisco. ISBN 0–7167–0464–1.

### BIOGRAPHICAL NOTE

*James Popple is a PhD student at the Department of Computer Science, Australian National University. He has degrees in law and computer science, and is interested in the areas where those two disciplines overlap: computer applications in the law, and the legal implications of computer technology. Mr Popple’s doctoral thesis research focusses on the development of legal expert systems.*