A CRITICAL REVIEW OF THE ATTEMPTS TO REPRESENT THE LAW AND THE LEGAL REASONING PROCESS IN COMPUTER PROGRAMMES

Cyril H. Ponnu
Faculty of Economics & Administration
University of Malaya
50603 Kuala Lumpur
Malaysia
Tel.: 603-7593682
Fax.: 603-7567252
email: g5cyril@cc.um.edu.my

ABSTRACT

Evaluates some of the previous attempts to represent the law and the legal reasoning process in computer programmes. It examines previous work in the area to demonstrate that researchers have generally ignored the complexity of the law and the legal reasoning process and have tended to focus on very narrow aspects of the law and legal reasoning process. The aim here is to highlight the limitations of the various approaches given that the nature of law and legal reasoning is such that it presents severe obstacles to those who undertake to represent the law and legal reasoning within a computer programme.

Keywords: ***Law, Legal Reasoning, Flowcharts, Decision Nets, Legal Expert Systems, Case-Based Systems***

1.0 INTRODUCTION

In general, previous attempts to represent law and legal reasoning in computer programmes have failed to appreciate the unique character of the law and the legal reasoning process. These attempts largely concentrated on extracting legal rules derived from statutes or case-law and representing it in a computer programme in a logical fashion. This is due, in part, to the failure of researchers to appreciate the nature of the law and the legal reasoning process, and in part due to the limitations imposed by the von Neuman architecture which makes it necessary for computer programmes to be written in a logical step-by-step procedure.

Generally, the focus of previous work have been to allow the facts of an actual or hypothetical case to be represented at some appropriate level of abstraction; it incorporates a representation of the law in one form or another; it is designed to 'apply' or 'match' this representation to the given facts to ascertain what legal concepts and relationships would seem to hold and what legal consequences would seem to follow.

But legal philosophy tells us that the law is more than that. Legal rules do not dictate legal outcomes; there is more going on in a case than simple deduction. The law has an added element in that there is often room for choice about which of several possible decisions is to be preferred. Another is that the choice, once made, sets a precedent, which may change the space of choices available in later cases.

Most current applications of computers to law fall outside the scope of this paper. For instance, word processing, court administration systems, legal document retrieval systems are excluded from the discussion here since these applications are not concerned with representing the law or the legal reasoning process.

Our main concern here is with systems that represent the law and the legal reasoning process. Due to the substantial amount of work done in this area, I have selected for consideration certain projects which are representative of a particular approach.

2.0 LOGIC-BASED APPROACH

The logic-based approach is a general approach which involves the construction of declarative representation of law using some convenient form of symbolic logic and the execution of these representations by automated theorem-provers, potentially for a variety of different purposes. The law is modelled by a set of logic sentences which represent some chosen unambiguous interpretation of the selected legal sources. Most applications of this approach have incorporated representations from statues only and have ignored other factors that might influence the outcome of a legal decision. The law is applied to the facts by deduction. This approach is similar to the algorithmic approach to representing the law. Sergot [1] argues that many kinds of legislation are essentially definitional in nature, and that this approach provides a simple and natural formal language for expressing such definitions precisely.

2.1 Formalisation of The British Nationality Act 1981

The best-known example of this approach is the representation of the British Nationality Act 1981 by Sergot *et al* [2] at Imperial College, London. It was implemented in APES which is an augmented form of PROLOG. This program is used to determine, for example, whether a given individual is a British citizen according to the provisions of the Act.

These logic-based formalism cannot adequately describe the process of legal reasoning required to resolve the disputes in question. The logic-based formalism breaks down when applied to cases involving the existence of conflicting rules and precedents; the ambiguity and imprecision of natural language on which most of the law depends; and the need for evolution in the legal system to adapt to a changing world. Lawyers, using accepted methods of legal analysis, can reach differing legal conclusions, a result not permitted in formal logic [3].

The inadequacy of formal logic in resolving legal reasoning can be explained by a number of reasons. First, logic assumes that each proposition is either true or false, with no middle ground. On the other hand, legal decisions often reflect a comprise between competing social, economic and political values. Second, formal logic is 'monotonic', that is, once a result flows logically from prior premises new information cannot be introduced that would justify a different conclusion. Thus, unlike the law that is always having to adjust to unusual cases, formal logic does not have any mechanism for adjusting to the exceptional cases. Third, logic does not permit meta-rules, that is, statements about the system's own rules. But throughout legal literature one finds jurists employing meta-rules to reach conclusions opposite from the results that would follow logically from strict adherence to the system's own rules.

Since symbolic logic applies rules to specific situations, several researchers have attempted to model legal concepts using the tools of formal logic. Rather than trying to model the entirety of legal reasoning, these experts have explored some rather narrow legal domains. Although these efforts may yield some systems that have some limited utility, they show that formal logic, though ideally situated for modelling mathematical systems, is less than ideal tool for replicating legal reasoning [4].

Leith [5] vehemently objects to the use of logic programming techniques in representing law. He argues that basing a computer system on 'paper rules' alone is equivalent to claiming that legal problems are resolved by mechanically applying 'clear legal rules'. As Leith points out, such rules are fictional. There is more to legal decision-making than applying the R x F = D equation, because the sources of law support different interpretations, and are subject to unstated exceptions, and are open to judicial manipulation, and because there are many other factors which need to considered besides legal rules. According to Leith, to consider only the 'paper rules' is tantamount to a fundamental misunderstanding of the true nature of law.

3.0 THE ALGORITHMIC APPROACH

This approach attempts to represent a part of the law by means of an algorithmic programming language such as BASIC, or a formalism, like flowcharts, which can be directly implemented in such a language. These systems normally operate by asking the user a series of questions until it has sufficient information to carry out its analysis [6].

3.1 Algorithmic Programmes

The most representative examples of this approach are by Hellawell dealing with certain aspects of tax legislation. They are all implemented in BASIC and they are CORPTAX [7], CHOOSE [8] and SEARCH [9]. As Hellawell himself admits, the difficulty in writing such a program lies in decomposing the legislation into such a minute detail so that it can be formalised in the required form. It requires the author to foresee all possible answers to all possible questions and provide an additional branch within the program for every combination of answers. Hellawell used flowcharts to determine which branch of the flowchart is followed.

3.2 Flowcharts

Twining *et al* [10] propose flowcharts as a means of understanding the intended effects of a complex statute. As was mentioned earlier, Hellawell used flowcharts in designing his programmes and it will be seen below that Greenleaf *et al* [11] also used flowcharts in their project. Thus, it can be said that flowcharts are an important feature in the algorithmic approach to representing law in computer programmes.

3.3 Decision Nets

The Data Lex project by Greenleaf *et al* use a programming shell called LES to build applications in law. This project, which makes extensive use of flowcharts, allows the user to build what are called decision nets. These nets are executed by an inference mechanism consisting of explicit interconnections between the modules which are activated by responses from the user. The LES shell translates decision nets into a C program which is then complied to produce an executable program.

The examples mentioned above are representative of the algorithmic approach to the representation of law. Their characteristic feature is that they represent legal provisions implicitly. Normally, a statute, or a part of it, is represented by an algorithmic which is designed to apply the statute's provision to the facts of given case in a certain, very specific way. Thus, this approach fails to take into account the open-texturedness of the law which is so fundamental to the nature of the law. The algorithm specifies just one particular way to interpret the legislation involved. Furthermore, algorithms also lack transparency [12].

4.0 CONCEPTUAL MODELS AND DOCU-MENT RETRIEVAL

In this section I will describe three projects which have attempted to improve the performance of document retrieval systems by incorporating a conceptual model of the underlying legal domain. Presently, document retrieval systems such as LEXIS and WESTLAW use text searches that are specified by Boolean combinations of key words. This systems are, therefore, necessarily limited. Thus, retrieval based on the meaning and contents of documents would be far more useful. Conceptual retrieval systems attempt to store and retrieve documents using an index based on a conceptual model of the particular legal domain.

4.1 Hafner's work

Hafner [13] used this approach to store and retrieve documents related to parts of the US Uniform Commercial Code. This legislation governs 'negotiable instruments' such as cheques and notes. Hafner constructed a database containing two hundred subsections of the code. The conceptual model allows the system to retrieve relevant documents even if none of the terms specified in the query appear in the document. Although the potential of Hafner's system appears to be great, the practical implementations remain to be investigated and built. As Hafner herself points out, this system involves a massive amount of work and poses a major practical obstacle.

4.2 CCLIPS

The CCLIPS (Civil Code Legal Information Processing System) project by Cross *et al* [14] aims to build a conceptual model of certain sections of the Louisiana Civil Code dealing with contract law. One of the main purposes of the CCLIPS project is to clarify the structure of the Civil Code itself. Once the conceptual model has been formulated, another aim of the project is to experiment with alternative designs for a conceptual retrieval system.

The formulation of the conceptual model is heavily influenced by Allen's normalisation process. Legal rules in the model are represented in the form prepositional rules. These rules are in turn expressed in terms of abstract primitive objects and abstract primitive actions. The aim is to capture the 'deep' semantic structure for the objects and actions that are encountered in the domain of contract law. The representation language is called Atomically Normalised Form (ANF) and the representations constructed are called Analytical Legal Database. The representations are intended to support the conceptual retrieval of legal information, but the system can also generate inferences. The facts of a case can be described in the ANF language in terms of the underlying conceptual model, and the rules can then be applied to this representation to determine legal consequences.

4.3 ESPLEX

The ESPLEX system under Biagioli *et al*.[15] is being developed as part of the Automated Analysis of Legislation project which has been conducted at the Istituto per la Documentazione Giuridica (IDG) in Florence since 1981. ESPLEX combines the use of logic for the representation of rules embodied in legislation with the use of frames and semantic networks to model the general legal concepts that are required to understand the legislation. The experimental domain for the project is the law relating to agricultural tenancies, part of the law of contracts. A main objective of the ESPLEX project is to develop a methodology for the analysis of legislation which takes into account the different kind of sentences and concepts in legal rules. The implementation language used is PROLOG.

5.0 CASE-BASED REASONING SYSTEMS

So far I have considered only statute-based systems. Now I shall consider a case-based reasoning system.

5.1 The HYPO System

The HYPO system developed by Rissland *et al* [16] for legal planning based on case-law in the domain of trade secrets law. The user inputs a fact situation which is analysed by the system. Then, the system retrieves relevant cases from a case-base and places the retrieved cases with respect to the case at hand. Finally, it selects the most pertinent cases, suggests relevant hypothetical cases, proposes a skeleton of an argument and justifies the argument with case-citations. The actual decision-making is left to the user, the function of the system is to generate arguments to assist in reaching a decision.

HYPO retrieves relevant cases by maintaining a library of dimensions. A dimension is a cluster of factual predicates having legal relevance for a particular claim in the domain of trade secrets law. That is, it corresponds roughly to one of the 'factors' that determine the outcome of the case. In addition, each of the dimensions contains a set of conditions which specifies when the dimension applies in a given case. These conditions are expressed in terms of factual predicates which are used to describe a case. Given a case described with factual predicates, the system's case-analysis module can thus determine which of the dimensions in the library apply to the facts of the case.

HYPO then uses the retrieved cases to construct the skeleton of a legal argument by citing cases for and against a particular decision in the new case. The argument is then refined by addition of hypothetical cases which are generated by applying a set of heuristics.

In contrast to some of the other systems mentioned in this chapter, the HYPO system recognises the adversarial nature of reasoning in law as it stress the generation of arguments. In this respect, it attempts to represent more of the true nature of the legal reasoning process. Furthermore, the use of dimensions, that is, recognising the importance of several facts combining to determine the truth-value of a certain legal concept, represents an important aspect of legal reasoning. The use of dimensions also ensures that the retrieved cases are legally relevant. However, as the system builders themselves recognise, legal dimensions do not act independently, and a model based only on dimensions is inadequate.

HYPO retrieves the most relevant cases which has the most dimensions in common with the dimensions of the case at hand. This kind of retrieval undoubtedly represents one approach used by lawyers in legal reasoning. However, there are other methods employed by lawyers in using cases other than this method. A case with just one fact in common with the case at hand may sometimes be more convincing than a case with many such common dimensions.

6.0 LEGAL EXPERT SYSTEMS

Here I will describe three examples of systems which can strictly be called legal expert systems.

6.1 Legal Decision Making System (LDS)

The LDS is a legal expert system developed by Peterson *et al* [17] in the area of settling claims in product liability cases. The system determines the defendant's liability and then calculates the amount that would be acceptable to all parties as a fair settlement. That is, it attempts to model how such claims are settled in practice out of court by stimulating the behaviour of lawyers and claim adjusters. LDS is constructed in the programming language ROSIE and the domain knowledge was elicited from human experts.

6.2 TAXADVISOR

A similar system to the one described above is the TAXADVISOR developed by Michaelsen [18]. This system is implemented using EMYCIN. It was developed for the purpose of assisting lawyers with tax-planning for their clients. The knowledge- base contains the kind of expert knowledge that is used by human tax-advisors. TAXADVISOR, like LDS, is another example of a system that has attempted to model, not so much the law itself, but the knowledge of how the law is actually used in practice.

6.3 Latent Damage System

This is a commercial legal expert system that advises lawyers on the application of the Latent Damage Act 1986. Developed by Capper *et al* [19], it addresses the difficult issue relating to the time within which a claimant may begin proceedings in the law of negligence if the loss or damage is discovered some time after its occurrence. It is an expert system because it incorporates Capper's expert analysis of the implications of the Act. However, it does not seek to represent what actually happens in legal practice or stimulate an expert's advice-giving behaviour.

The limitations of the above systems are that they tend to restrict attention to a form of legal analysis which resembles R x F = D. They do not consider the benefit of incorporating the social setting that the law functions in. Rather, only 'paper rules' of law are considered. But in comparison with other approaches, the above systems, particularly LDS and TAXADVISOR tends to incorporate more of a lawyer's actual reasoning process.

7.0 THE NATURE OF LAW AND LEGAL REASONING

However, the dynamic nature of the law and the legal reasoning process renders the kind of characterisation we saw above superficial. As Gardner [20] comments,

> "In most previous computational work on legal reasoning, legal issues have been treated as if they are all alike. In some work the program is presented with a case raising a single major issue, which is assumed not to have a clear-cut answer, and is expected by some means to weigh the factors favouring a decision either way. In other work it is assumed that deduction from the rules is sufficient, perhaps supplemented by asking the user what choice would be made on some points. That the user might not know, and may wrongly assume that he or she

> does not know, is usually not taken into full account"

Walter [21], reflecting upon the current attempts to represent the law in computer programmes, comments,

> "Like many other aspects of human activity, the law is what it is. Making it to fit the limitations of existing technology may provide theorists with valuable learning experiences about law and the technology employed, but it won't provide legal practitioners with the tools they need."

According to Leith [22] the task of representing law in a computer program is even more complex than in other fields because it is one of the most complex social science fields.

> "The law is where social control, power and diverse amounts of wealth meet. These are not facets which can be easily reduced to algorithm and data which can be incorporated in a program. The law must be seen in its social context if it is to be successfully represented in a computer program."

All too often, the complexities of the legal reasoning process is overlooked when attempting to represent the law in computer programmes. Computer applications in the law requires understanding the non-technical as much as the technical aspects. Leith [23], mentions as an example, the development of logic programmes which supposedly can give information on who is a British citizen, can unwittingly, lead to the formalisation of the frequently found racism inherent in that legislation and its judicial interpretation. Thus, computer science is in error when it implicitly assumes that the difficult issues are only found in technical problems. As Leith [24], eloquently puts it,

> ".....it is technalism which suggest the same techniques which can be used to provide expert systems in areas such as combining boards in a computer's circuitry are appropriate for decision making in business, or the interpretation of statutes. This is an errant view, because it is not simply that business or law are more complex than computer configuration. Rather it is that they are qualitatively different."

Elias [25], echoes the same sentiment when he points out, that scientific, engineering and similar fields are characterised by a high degree of detachment and general agreement of perspective about what the discipline is concerned with. On the other hand, in the domain of law, like the other areas of social sciences, this detachment is missing. There is much more involvement of the researcher in the description of the problem, which often indicates the particular class perspective of the researcher. Thus, when we develop models there is a bias which exists and which is derived from our own perspectives within the system.

Specifically referring to the domain of law, Leith [24] suggests that,

> "As regards models of law, it can clearly be seen that individual perspectives of the existence or otherwise of social cohesiveness bears upon the different models of law - the ruled-based view of H.L.A. Hart demonstrating the bias of belief in society's commitment to the particular rules in evidence at any one in time. Such a bias can be categorised as particularly bourgeois. The very nature of 'the problem' to be solved by computerisation is thus biased, for in the Hartian view it becomes the technical problem of handling these socially accepted rules.
>
> In a different model of law which is biased more towards the view that rules are imposed (i.e. that the law is, in part at least, a dissentual system) the problem to be solved by computerisation is most certainly not the simple handling of rules....we can describe the problem....as having a 'subjective objective'."

8.0 CONCLUSION

I have looked at some of the problems relating to representing law and legal reasoning in computer programmes. I also considered some of the jurisprudential issues relating to the nature of law and legal reasoning and their implications for computer representation. It was pointed out that the peculiar nature of law presents obstacles to those who attempt to represent the law and legal reasoning in computer systems. I also critically evaluated some of the previous work in this field to demonstrate that many applications have either ignored or failed to appreciate the unique nature of law and legal reasoning.

This paper shows that many of the applications have chosen to represent only a small part of the legal reasoning process and are thus of little practical use to the lawyer or even the layman. Thus, the hard question now facing researchers is to appreciate better the character of the law and legal reasoning and to develop technology capable of representing substantially more of the richness of the law than current technology is capable of doing.

REFERENCES

[1] M. J. Sergot, "Representing Legislation As Logic Programmes". *Machine Intelligence, 11, August. 1985*.

[2] M. J. Sergot, H. T. Cory, P. Hammond, R. A. Kowalski, F. Kriwaczek, F. Sadri, "Formalisation of the British Nationality Act". *Yearbook of Law, Computers and Technology, Vol. 2.* Butterworths, London, 1986.

[3] D. Berman, & C. Hafner, Obstacles to the Development of Logic-Based Models of Legal Reasoning. In C. Walter (ed.), *Computer Power and Legal Language: The Use of Computational Linguistics, Artificial Intelligence and Expert Systems in Law.* Quorum Books, pp. 184-214, 1988.

[4] P. Leith, "Fundamental Errors in Legal Logic Programming". *The Computer Journal, Vol. 29, No. 6, 1986.*

[5] P. Leith, "Legal Expert Systems: Misunderstanding the Legal Process". *Computers and Law, No. 49, Sept. 1986.*

[6] M. Sergot, (1991), "The Representation of Law in Computer Programmes". In T. J. M. Bench-Capon (ed.), *Knowledge-Based Systems and Legal Applications.* Academic Press, London, 1991.

[7] R. Hellawell, "A Computer Program for Legal Planning and Analysis: Taxation of Stock Redemption". *Columbia Law Review 80, 7,* pp. 1363-1398, 1980.

[8] R. Hellawell, "CHOOSE: A Computer Program for Legal Planning and Analysis". *Columbia Journal of Transnational Law 19*, pp. 339-357, 1981.

[9] R. Hellawell, "SEARCH: A Computer Program for Legal Problem Solving". *Akron Law Review 15*, pp. 635-653, 1982.

[10] W. L. Twining, & D. Miers, *How To Do Things With Rules.* 2nd Edition, Weidenfeld & Nicolson, London, 1982.

[11] G. Greenleaf, A. Mowbray, A. L. Tyree, "Expert Systems in Law: The DATALEX Project". *Proceedings of the First International Conference on Artificial Intelligence and Law,* ACM Press Boston, 27-29 May, 1987.

[12] M. Sergot, The Representation of Law in Computer Programmes. In T. J. M. Bench-Capon (ed.), *Knowledge-Based Systems and Legal Applications.* Academic Press, London, 1991.

[13] C. D. Hafner, Representation of Knowledge in a Legal Information Retrieval System. *Information Retrieval Research*. Butterworths, London, 1981.

[14] G. R. Cross & C. G. deBessonet, The Implementation of CCLIPS. In C. Walter (ed.), *Computer Power and Legal Language: The Use of Computational Linguistics, Artificial Intelligence and Expert Systems in Law.* Quorum Books, New York, pp. 89-98, 1985.

[15] C. Biagioli, P. Mariani, & D. Tiscornia, ESPLEX: "A Rule and Conceptual Based Model for Representing Statutes", *Proceedings of the First International Conference on Artificial Intelligence and Law, Boston, 27-29 May 1987*, pp. 240-251, 1987.

[16] E. L. Rissland, & K. D. Ashley, "A Case-Based System for Trade Secrets Law". *Proceedings of the First International Conference on Artificial Intelligence and Law, Boston, 27-29 May 1987*, pp. 60-66.

[17] M. A. Peterson, & D. A. Waterman, Evaluating Civil Claims: An Expert System Approach to Evaluating Product Liability Cases. In C. Walter, (ed.), *Computing Power and Legal Reasoning.* West Publishing Co., St. Paul, Minn., pp. 627-659, 1985.

[18] R. H. Michaelsen, "An Expert System for Federal Tax Planning". *Expert Systems 1,* p. 2, 1984.

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

[20] A. V. D. L. Gardner, *Artificial Intelligence Approach to Legal Reasoning.* MIT Press, p. 189, 1987.

[21] C. Walter, (1988), Expert Systems in Law. In C. Walter (ed.), *Computer Power and Legal Language: The Use of Computational Linguistics, Artificial Intelligence and Expert Systems in the Law*. Quorum Books, pp. 330-349, 1988.

[22] P. Leith, *The Computerised Lawyer: A Guide to the Use of Computers in the Legal Profession.* Springer-Verlag, p. 201, 1991.

[23] P. Leith, *Formalism in AI and Computer Science.* Ellis Horwood, Chichester, 1990.

[24] P. Leith, *The Computerised Lawyer: A Guide to the Use of Computers in the Legal Profession.* Springer-Verlag, pp 203, 1991.

[25] N. Elias, *Involvement and Detachment*. Basil Blackwell, Oxford, 1987.

BIOGRAPHY

Cyril H. Ponnu obtained his B.Ec from the University of Malaya in 1980, his LLB form the University of Wolverhampton in 1992 and his M.Sc in Management Science from the University of Southampton in 1993. Currently, he is a Lecturer in Business Law and Ethics in the Division of Business Administration, Faculty of Economics & Administration, University of Malaya. His research interests include the application of computers to law, computer ethics and artificial intelligence.