BUILDING INTELLIGENT LEGAL TOOLS - The IKBALS Project

John Zeleznikow
Database Research Laboratory
Applied Computing Research Institute
La Trobe University
Bundoora Victoria Australia 3083

FAX: 61.3.470-4915

ABSTRACT:

This paper discusses the construction of intelligent legal tools. It argues that such tools should provide litigation support, rather than be judgement systems. It recommends that such systems should use both statutes and precedents. The paper then suggests implementing such systems using both rule-based and case-based reasoning. It is also suggested that tribunal based systems are ideal for such modelling. A detailed analysis of a prototype (IKBALS) which gives advice on the Accident Compensation Act (1989), Victoria, Australia, follows. The paper concludes by describing future research projects.

KEYWORDS: Legal Expert Systems, Litigation Support Tools, Rule Based Reasoning, Case Based Reasoning, Accident Compensation Act, Tribunal Based Systems.

I. INTRODUCTION

Computers are widely used in the legal profession, assisting lawyers, judges and the courts with the day to day administration of justice [Vossos et. al., 1991a]. Until recently, the major use of computers in litigation support was that of 'Legal Information Retrieval Systems'. Such systems are usually on-line databases containing legislation as well as past cases.

While such computerized tools can be helpful to lawyers for the retrieval of legal material, many believe that they are not very useful in the litigation support role [Susskind, 1987]. Recently, there has been an acceptance of the need to develop computer systems that display intelligent legal reasoning.

1.1. Automating Legal Reasoning

Legal reasoning can be viewed as an intellectual process by which lawyers and judges use rules (statutes or regulations) and previously tried cases (precedents) to solve legal problems. Legal practitioners primarily combine two forms of reasoning when dealing with litigation: reasoning by deduction and reasoning by analogy [Levi, 1948], [Burton, 1985]. Legal reasoning is more than 'deduction', whereby lawyers rely on annotations and explanatory material (reports, practice guides, precedent cases, opinions of academic and distinguished lawyers) to help add some contextual information to legal rules. Hence, legal reasoning can be viewed as an attempt to interpret statutes initially through the use of the rules, referencing precedent cases only when the rules run out, or when the use of rules prove insufficient in elicitating concepts.

It is also our belief, that such systems should function as decision support systems, which help lawyers retrieve relevant legal information. This is in contrast to the work of [Sergot et. al., 1986] who built a legal expert system to determine eligibility for British Citizenship. The problems with viewing legal reasoning as a 'system of rules' has been well documented [Hafner & Berman, 1990], [Tyree et al, 1987], [Gardner, 1987]. Besides the 'syntactic difficulties' inherent in normalizing statutes in the form of rules, there are 'semantic' difficulties such as the presence of conflicting rules, imprecise terms and incompleteness. The goal of this project is to design a computer model which will be capable of 'lending a hand' to a rule-based reasoner when it encounters open-textured legal predicates which it cannot resolve given the rule set(s) available and the facts of the current case.

Because, the law is adversarial in nature, legal practitioners are in general not solely interested in the likely outcome of a court case. They are concerned with providing support to argue the case that their client wishes to put forward, although admittedly they do encourage their client to pursue a path that has a reasonable chance of success. Hence what is really needed is a litigation support system which will allow the lawyer to navigate through the vast amount of legal sources available, permitting him to find the relevant rules and precedent cases to successfully argue his client's case. It is important that the system make the lawyer aware of precedents supporting his opponent's case.

In this paper, we discuss how to construct intelligent legal tools, rather than building judgement systems. To do so, we need to extend the early if-then production rule systems of [Waterman, Paul and Peterson, 1986] and [Schlobohm and Waterman, 1987]. In particular, we need to introduce the concept of case based reasoning [Riesbeck and Schank, 1989], which will allow us to reason with precedents. To implement case based reasoning, we shall need to use the object oriented paradigm [Thomas, 1989].

We shall also discuss which statutes are ideal for modelling, and why we chose the Accident Compensation Act (1989), Victoria, Australia as the legislation to model. We shall argue that intelligent legal tools can effectively model tribunal based systems, and that our Prototype, IKBALS II can be extended to model most tribunal based legal systems.

We shall discuss the basic features of IKBALS. The reader is referred to [Vossos et. al., 1991a] and [Vossos et. al., 1991b] for full technical details of case based reasoning, the object-oriented paradigm and the IKBALS system. It should be noted that because legal decision making generally involves a dispute over the interpretation of facts, it is futile to build legal tools for laypeople. IKBALS helps lawyers with little knowledge of the act give advice to their clients re entitlements under the act.

We conclude by discussing extensions to IKBALS II, as well as other legal expert systems we are currently constructing.

1.2. Knowledge Engineering

Knowledge Engineers, try to structure specific knowledge in a given domain, into a schema/representation which can be efficiently accessed and manipulated. In many ways they can be compared to barristers, who are specialists in advocacy. Barristers quickly learn the facts and jargon of the case they need to present. They must rely on expert witnesses for their understanding of the case. In a similar manner, knowledge engineers must rely on the domain expert (namely lawyers) when building legal expert systems.

In the Database Research Laboratory, Applied Computing Research Institute, La Trobe University, we are trying to develop concepts and software tools specifically designed to aid practitioners to reason with information in domains characterised by large volumes of qualitative data [see Yuen et. al. (1991)]. In such domains, it is rare to find solutions to problems by simply applying algorithms or invoking deductive rules in some knowledge based program. Instead, expert practitioners often supplement domain specific deductive knowledge by experience. This type of expertise is often applied in the form of an analogy [Ashley (1988)]. Our work models analogical reasoning in a variety of task orientations including organisational planning (DORAS), legal reasoning (IKBALS) and medical diagnostic reasoning (FLORENCE). In all cases involving precedents or experience, Case Based Reasoning (CBR) has been chosen as the methodology for realising the automation of analogical reasoning. We aim to generalise our findings and to develop a formal methodology for building case based reasoners. Such case based reasoners are essential for the development of intelligent legal tools.

We have chosen to develop our models using some of the more advanced features of existing knowledge engineering environments. These techniques allow us to extend current object-oriented/rule based architectures to handle the subtleties of case based reasoning. This is achieved by:

(i) integrating with other technologies such as hypermedia and model based reasoning [Davis and Hamscher,1988], [Rissland and Skalak, 1990].

(ii) distributing the problem to be worked on over a network of individual autonomous agents.

IKBALS II [Vossos et. al.,1991b] relies on a hypertext engine for its text representation of the actual cases and statutes as well as its user interface.

It should be stressed that IKBALSII is a research prototype, not a commercially viable robust legal expert system. Since it is our aim to develop intelligent legal tools, we are concentrating on developing new systems. We accept the arguments of [Moles,1991] that current legal expert systems inadequately model the legal process. We are hence concentrating on building fundamentally new systems.

2. MODELLING LEGAL REASONING

Until recently, most legal knowledge based systems have generally tried to model the problem of statutory interpretation by simply normalizing sections of the legislation in the form of 'if_then_' rules, and then applying control and heuristic information in order to efficiently guide the logic of the system; (programs=logic + control [Kowalski,1979]). In consequence, the knowledge in these early legal knowledge based systems amounted to nothing more than collection of rules reliant on static necessary and sufficient conditions and meta-rules (heuristic information used to control the system) chained together in order to determine if a current case could be classified as belonging to a particular legal category [Skalak, 1989]. Once again, the bulk of the knowledge base was composed of 'control' or meta-rules rather than domain-specific rules. Such deductive reasoning allowed rules to be either 'forward chained': considered analogous to firing sufficient conditions, and/or 'backward chained': analogous to firing necessary conditions.

Although such rule-based models are both theoretically and practically very powerful, their suitability in reasoning with open textured domains such as law, however, must be doubted, especially considering their inefficiency in adequately capturing contextual information contained in previously tried cases which must be considered when reasoning about the merits of a current case.

From a technical perspective, representing legal rules in a strictly rule-based system is prone to the following limitations if sophisticated control techniques are not incorporated [Vossos et. al., 1991b]:

1. Inefficiency - The inference engine may have to search through a very large number of rules at each stage of execution. This results in a high problem solving overhead;

2. Maintainability - A disadvantage associated with rule based systems is that it is hard to follow their flow of control;

3. They may not properly reflect the reasoning structure used by the expert;

4. The open textured nature of legal knowledge and reasoning.

The last two factors often result in the knowledge mis-match problem which leads to cumbersome representations and laborious reasoning to reach decisions, which may not be acceptable.

Lawyers when considering a particular dispute normally try to find all the relevant cases to that particular dispute. They extract the applicable principles or interpretations from these previous precedent cases in order to form a judgement as to whether to proceed with the dispute. If so, they develop an argument in support of their position in the dispute, citing appropriate sections of the legislation and appropriate cases. They search for arguments which indicate that the precedents supporting their case are relevant to the particular case, whilst arguments supporting the opposition case are not.

Some means of organizing the overwhelming mass of legal material a lawyer must process when reasoning about a case is essential. As discussed previously, it is difficult to express such knowledge simply as production rules. It appears therefore that in modelling legal reasoning, one requires the doctrines of the law; the particular statutes; and the legal arguments and particular facts and circumstances contained in precedents.

Whilst there might be some chance that rule based or logic based systems could capture some of the elements of the statutes, it is highly unlikely that they will be able to capture the remaining requirements. This has led to our interest in IKBALS II as a means of dealing with the subtlety and complexity inherent in legal reasoning problems by experimenting with both deductive and case based reasoning.

2.1. Choosing Legal Systems to model

In a number of areas of the law in Australia and its states, there are alternatives to the courts which help reduce the number of cases litigated. These can be considered as Arbitration/Tribunal systems. Some examples include the Conciliation and Arbitration Commission (to mediate on industrial disputes and awards), the Small Claims Tribunal (to handle litigation over small amounts of money concerning consumers), the Town Planning Appeals Tribunal, and the Accident Compensation Tribunal. These are often staffed by expert laypeople (possibly lawyers) who are not necessarily judges. Such tribunals generally act more quickly than and are cheaper than resorting to the courts. Their decisions can be appealed to the courts. In general, tribunals help reduce costs, aid in alleviating the backlog of court cases, and encourage mediation rather than litigation between the feuding parties.

It is useful to classify the judicial system into two separate sections for the purpose of automation, namely Tribunal judicial systems and Court Based systems. Tribunal systems do not follow the laws of evidence as strictly as court based systems. In addition, tribunal systems allow mixed questions of law and fact to be discussed, and it is common for non-lawyers to appear in this jurisdiction. Whilst court based systems must treat precedents in a formal manner, tribunal systems are not required to do so. Whilst the legal principles underlying tribunal and court based systems are the same, the practice of law in them is very different.

The knowledge that is utilized by Tribunals can be characterized to involve a large number of statutes and regulations. Further, the knowledge utilized in the decision making processes is primarily hierarchical and is often difficult to maintain as legislation is prone to change.

Nevertheless, because the legislation regularly changes, lawyers are often unfamiliar with the new statutes and precedents. They hence find such Legal Expert Systems invaluable. Thus even though maintaining Legal Expert Systems can be very difficult and costly, the changing nature of legislation makes the provision of Legal Expert Systems such a valuable tool for lawyers. Rather than spending many hours in libraries, and consulting their colleagues about the new legislation, they can consult the relevant legal expert system.

2.2. The Accident Compensation Act (1989)

It has been our desire to build a prototype system to provide intelligent legal advice. In choosing legislation to model, we took the following criteria into account:

1. The legislation needed to be new, so that there would not be a large precedent base;

2. The legislation needed to be tribunal based;

3. The judicial system giving rulings under the act should have very little discretionary power (unlike judges have under the Family Law Act);

4. The legislation should constantly change (hence making such a system invaluable for lawyers);

5. There needed to be much litigation under the act (ensuring the financial viability of building such systems).

IKBALS II deals with the statutory interpretation of the Accident Compensation (General Amendment) Act, 1989, (WorkCare). Emphasis is paid to unsuccessful claims which must be appealed to by the worker's council to the Workcare Appeals Board with a further right of appeal to the Full Bench of the Appeals Board and/or to the Accident Compensation Tribunal. In particular, the system focuses on elements giving rise to a WorkCare entitlement.

The Act gives a worker an entitlement to compensation if he suffers an injury arising out of or in the course of employment (S82). Unlike actions at common law, it is not necessary to show fault (negligence). It is however, crucial to the success of any claim that the applicant falls within the following statutory definitions:

Worker- The Act extends the normally understood meaning of employer to include such people as contractors, owner/drivers and commissioned agents. Casual and part time workers are covered in the same way that full time workers are covered;

Injury- This means any physical or mental injury, including industrial deafness and diseases contracted in the course of employment. It also includes the recurrence, aggravation, acceleration, exacerbation or deterioration of any pre-existing injury where the worker's employment was a contributing factor;

Out of or in the course of employment- The 'work connection' is at the core of any claim for compensation. In general, a worker who suffers an injury in the following circumstances is protected by the Act:

1) the injury occurred at work or was caused by work;
2) the injury occurred in circumstances prescribed by s. 83 - deemed worker;
3) a disease caused, aggravated, accelerated, exacerbated or which was made to deteriorate or recur by employment.

In the case of a disputed claim, the lawyer first proceeds to prepare a written submission for the worker in order to present it before the Workcare Appeals Board. IKBALS II helps the lawyer evaluate the merits of a case by keeping an up to date case-base of previously tried cases, (precedents), in addition to expert domain specific knowledge. The organization of the structure of these sources of knowledge can be found in [Vossos et. al.,1991b]. Lawyers are able to investigate the likely outcome of their input case by comparing it against other previously tried cases which have come before the Appeals Board of the Tribunal by contrasting it against facts that the court identified as significant in determining the case. The system bases its comparison of the case on information obtained from the worker and medical practitioner concerning the circumstances of the injury and the degree of incapacity.

The Workcare legislation is relatively new and hence it is simple to maintain a complete and up-to-date precedent case base. Nevertheless the legislation has undergone many changes, and hence lawyers find it difficult to keep abreast of the amendments. IKBALS II is capable of helping alleviate many of the problems associated with frequent amendments.

Suppose that an injured worker walks into a solicitor's office requesting advice on the nature of his injury. The worker was injured while returning home from a union meeting. The meeting was originally scheduled on a Friday, but due to difficulties in hiring a hall the meeting was re-scheduled for Sunday. On his return home from the Sunday meeting, the worker was crossing the road when she was struck by a motor vehicle. The worker's doctor claims the worker will be incapacitated for at least twelve weeks.

The lawyer, not being too experienced with Workcare will use the system to recommend the best way to pursue the claim. We will refer to this current input case as [CIC] in the remainder of this paper.

IKBALS II reflects the expertise that goes into making these legal decisions for a wide variety of circumstances. It takes into account the nature of the injury, the degree of incapacity and identifies any work connection.

3. CASE BASED REASONING

Case based reasoning (CBR) is an approach to problem solving based on retrieving and applying stored solution examples (cases) [Ashley and Rissland,1988]. The basic idea of Case based reasoning is thus: A case based reasoner solves new problems by adapting solutions that were used to solve old problems. This differs from rule based areas.

This problem solving methodology brings up a variety of research issues, which we are currently addressing - Given a set of cases, how is the most relevant one selected? What happens if the chosen case fails to accomplish the goal? What knowledge is needed to adapt a case to a new problem? How should case memory be organised for efficient retrieval?

3.1. Legal Case Based Reasoners

Because of the deficiencies of traditional rule-based expert systems to successfully model the legal process, considerable interest has been focused on precedent case based reasoning. CBR uses past cases, or precedents, to find an interpretation of a current input case based on

(a) the point of view of the user and

(b) the intersection of 'similar' features occurring between the set of features present in the input case and the set(s) of features present in precedent cases. From a set of most relevant cases retrieved, a smaller subset of most promising cases(s) is selected by focusing on the importance of shared, relevant similarities. A case based reasoner would then proceed to justify the line of argument by explaining its interpretation.

Although CBR is still a research paradigm, it has been shown to offer significant advantages over conventional rule-based expert systems when attempting to reason with experience, or in our case, with previously tried cases. Research issues still confronting CBR include the representation of episodic knowledge, memory organization, indexing, case modification and learning [Slade, 1991]. Despite these technical issues, CBR has been moderately successful in dealing with the subtlety and complexity inherent in legal reasoning problems. The books of [Riesbeck and Schank, 1989] and [Kolodner, 1988] describe numerous legal case based reasoning systems. JUDGE [Bain, 1986], works in the domain of criminal sentencing by modelling a judge who is determining sentences for people convicted of crimes; whilst HYPO [Ashley & Rissland, 1988; Ashley, 1988], does case based reasoning in the area of patent law generating plausible arguments for the prosecution or the defence.

A legal case based reasoner essentially reasons from previously tried cases, comparing the contextual information in the current input case with that of cases previously tried and entered into the system. Legal classification of the current case is achieved by:

(i) analogising the facts and circumstances of the current case with those in the case knowledge base (CKB)

(ii) applying similarity metrics in order to retrieve the most 'on-point' cases. Precedent cases in the knowledge base are organized so that the features of the current case can be used to index them.

After the best match has been determined, the lawyer can then engage in constructing a sound case that he will present before the court in favour of his client. This is achieved by allowing the lawyer to strengthen his argument or weaken his opponent's argument by investigating the consequences of adding/subtracting or strengthening/weakening key attribute values of his input case that will effect the subset of applicable dimensions retrieved.

3.2. The Object Oriented Methodology in IKBALS

We present a brief review of the Object Oriented Paradigm to facilitate the discussion in the rest of the paper.

An object consists of an encapsulated representation (data structure) and a set of methods (operations or procedures) that can be applied to the object in order to activate it to do something. Encapsulation is the technical name for information hiding. Recent approaches permit methods to be collections of rules or rule sets

Only the methods of an object have access to its data structure, and a method can only be invoked by sending the object a message. Since a method is part of an object and not a global entity, there is no problem with two separate objects having a method by the same name. That is, sending the draw message to a Line object invokes its draw method; sending the same draw message to a Circle object invokes a different method. The ability of different objects to respond differently to the same message is known as polymorphism.

The other characteristic of object oriented programming is inheritance. This is the ability to define a new object that is just like an old one except for a few minor differences. Inheritance increases code sharing by allowing the language rather than the programmer to reuse code from one object (data structures and methods) in another related object.

Collections of objects which have similar properties can be grouped together into classes or subclasses. A subclass represents a specialization of a particular class. Subclasses and objects can inherit attributes and methods from superclasses.

The basic structure used for representing the knowledge in IKBALS is the object oriented approach employing a lattice of classes and objects. These lattices consist of classes or objects connected in a fashion which supports direct inheritance and multiple inheritance. These classes and objects have associated data structures and methods. The data structures essentially provide a mechanism for storage of attributes. In order to implement object oriented systems a frequent approach is to map the object oriented system onto a framebased structure underlying the representation.

Figure 1. is a CaseObject Diagram. All precedent cases in the Case Knowledge Base (see section 4.2.2.) are instantiated from the class ClassObject before being attached into the Case Knowledge Base Diagram.

fig. 1. below - CaseObject Diagram

4. *THE IKBALS SYSTEM*

4.1. IKBALS I

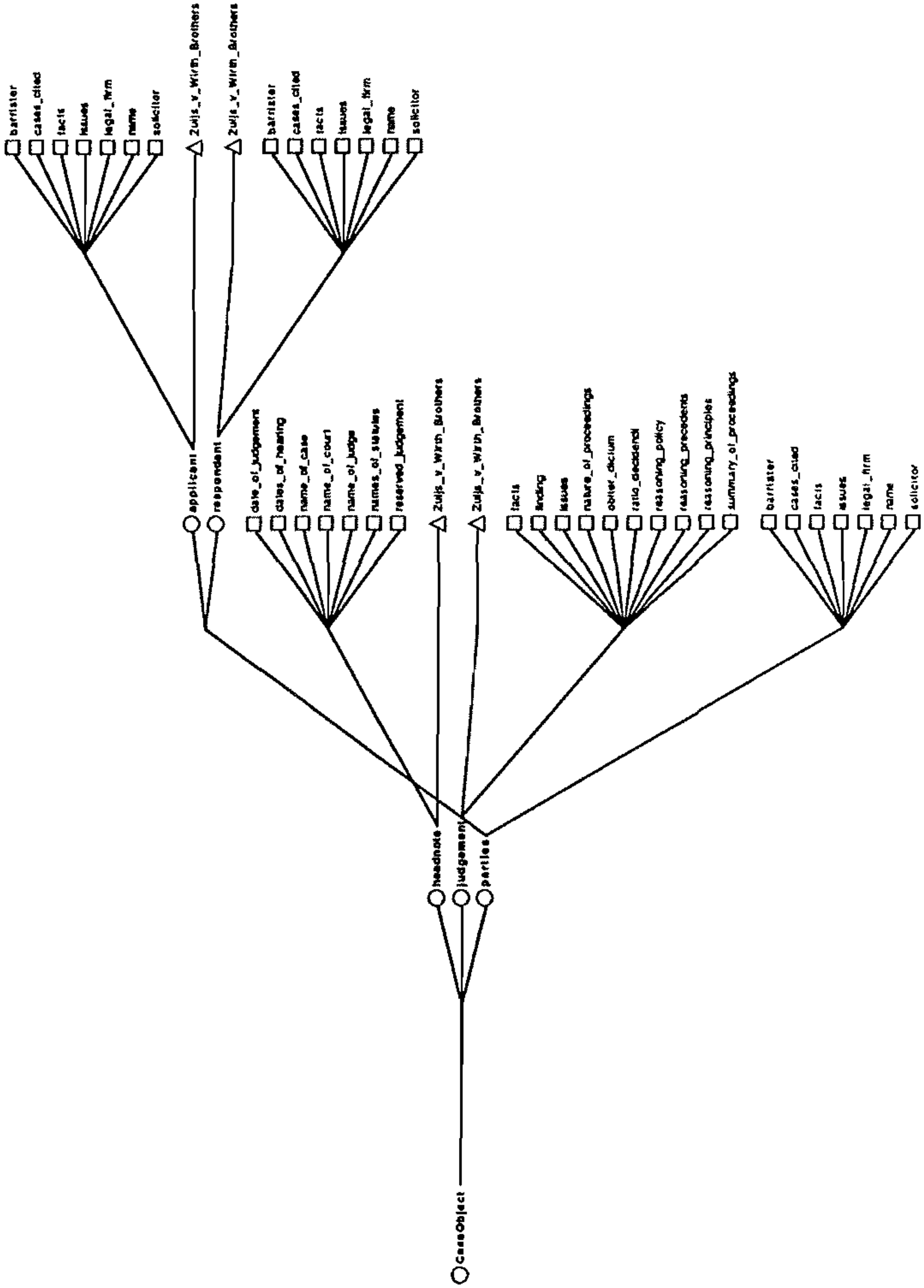
The original IKBALS system was developed on a 386/Windows Workstation running Goldworks II. GoldWorks II is an object oriented knowledge engineering environment that provides advanced capabilities for building knowledge based systems. The features of multiple inheritance, multi valued facets, goal directed forward chaining, bidirectional rules, confidence factors, direct control over the Agenda, and the dynamic graphics tool made it the prime candidate for building IKBALS.

IKBALS I (as it is now known) checks if the the client is eligible for relief under the act, and if so, proceeds to determine his eligibility for certain benefits. It should be noted that IKBALS I does not reason with precedents. The IKBALS I prototype was finished in November 1989.

4.2. IKBALS II

Given our desire to allow our hybrid/object oriented rule based system IKBALS I to reason with precedents, it became important that we add case based reasoning to IKBALS.

Rissland and Skalak are currently extending the HYPO CBR system [Ashley and Rissland, 1988] to incorporate rule based reasoning (CABARET) [Rissland and Skalak, 1989]. They use a mixed paradigm approach to the problem of statutory interpretation, combining case-based reasoning with rule-based reasoning (RBR). These two co-reasoners are supervised by their own dedicated monitors which make observations towards the progress of a solution. Observations are reported to a controller that uses the monitors' observations to decide on how i) the system as a whole is to proceed and ii) how the individual co-reasoners are to proceed using the controller's set of 'control heuristics'. The system has been implemented using a frame-based, 'blackboard' expert-system architecture.


barrister
cases_cited
facts
issues
legal_firm
name
solicitor
Zuijs_v_Wirth_Brothers
Zuijs_v_Wirth_Brothers
barrister
cases_cited
facts
issues
legal_firm
name
solicitor
applicant
respondent
date_of_judgement
dates_of_hearing
name_of_case
name_of_court
name_of_judge
names_of_statutes
reserved_judgement
Zuijs_v_Wirth_Brothers
Zuijs_v_Wirth_Brothers
facts
finding
issues
nature_of_proceedings
obiter_dictum
ratio_decidendi
reasoning_policy
reasoning_precedents
reasoning_principles
summary_of_proceedings
barrister
cases_cited
facts
issues
legal_firm
name
solicitor
headnote
judgement
parties
CaseObject

4.2.1. The IKBALS II Methodology

In the IKBALS project, CBR is not used in its most general form but merely to retrieve relevant cases when the rules are either inadequate or silent in their definition of WorkCare concepts (i.e., the employment relationship, injury or disease, and work related injury).

At first we made the naive assumption that legal reasoning can be considered as an attempt to interpret statutes initially through the use of the rules. If this is so, the basis of any such reasoning system must necessarily be rule/object based. Eventually rules must run out, otherwise the reasoning system would be infinite. It is precisely at this stage that we need to use precedents to guide the legal decision reasoning process, i.e., at the point where the rules themselves prove insufficient in their elicitation of concepts. We hence need techniques to retrieve the relevant cases once reasoning with the rules is exhausted or inconclusive.

It should be noted that any precedent base consists entirely of cases where the rules have proved inadequate. No case which is a clear interpretation of the statute would be litigated under the WorkCare system. Thus in some way, only the exceptional cases which go to litigation are recorded in the precedent base. Hence given the domain of application, the legal reasoning system must primarily be rule/object based. However to deal with certain situations, the system requires the efficient intelligent retrieval of relevant precedents.

In order to successfully combine CBR with the present rule-based architecture of IKBALS, it was important to structure the key elements of both the rules and cases so as to facilitate the efficient interleaving of these knowledge sources. Our technique involves the use of a lattice comprising of object/class structures used to represent both the taxonomic hierarchies that exist when reasoning in the WorkCare domain as well as providing an organization for the clusters of dimensions used to index the cases in the Case Knowledge Base (CKB). Intersecting decision sets, in the form of rule sets, are also used to represent the collection of legal heuristic knowledge needed to reason in the WorkCare domain. Such a representation leads to a more natural and elegant model of the legal reasoning process in our domain, as well as a large reduction in the number of cases that need to be examined in the Case Knowledge Base (CKB) for relevance during the matching process.

For a full discussion about the design of IKBALS, including the inferencing strategies, refer to [Vossos et. al., 1991b]. It should be noted that whilst CABARET uses a blackboard system, IKBALS II is considering storing cases as agents. Each case will store its fundamental characteristics in an agent, and the agents of each case will communicate with the agents of other "like" cases.

4.2.2. The IKBALS II System

IKBALS II relies on a hypertext engine for its text representation of the actual cases and statutes, as well as its user interface. Hypermedia is a very powerful information representation technology allowing the designer to represent information in a linear, hierarchical or arbitrary connected network. Because of the requirement for lawyers to be able to navigate through the vast amount of precedent cases and legislation, it was decided to use a hypertext engine to drive IKBALS II. This decision means that the lawyer can now browse through related pieces of information by pointing and clicking buttons or key words that appear in a text document, effectively linking him with sources of related information. The lawyer can then trace back through the sources of information (nodes) in order for him to get back to a previous reference. Hypercard v1.1.2 is currently being used for this purpose.

Apart from the representational advantages associated with modelling highly qualitative domains such as law with Hypertext, Hypertext systems also facilitate for rapid prototyping of the end-user system. This feature is tied in with the fact that some hypertext systems are object oriented, permitting developers to build a great deal of the system by 'pointing, clicking and dragging' icons on the screen. HyperCard allows us to quickly and efficiently design and build the user interface as well as some of the knowledge structures, in particular, the full text summaries of all the cases reported with their associated links to other cases and regulations. Nexpert v2 is currently being used as the knowledge engineering environment interfacing with HyperCard under Nexpert's HyperBridge.

IKBALS II works thus: The input to IKBALS II is a problem situation is entered via a series of templates (rather than in natural language). The system then proceeds to:

(a) identify the relevant legal norms;

(b) attempts to categorize the worker' claim for compensation under the Act by chaining backward and forward rules in order to satisfy the necessary and sufficient conditions required to satisfy the legal norms;

(c) identify relevant precedent cases for concepts or terms that cannot be resolved given the facts of the case and the rule set(s) available, in a reliable efficient manner, from a very large number of possible cases;

(d) having identified the relevant precedent cases, it proceeds to compare it with other cases in the system, and in particular, distinguish the current case from others whose conclusions are to the contrary;

(e) suggest arguments which can be made in favour of the current case and the facts and precedents which can be cited to support them.

IKBALS II represents three types of domain knowledge that allow it to perform this kind of CBR:

- the legal precedents which are represented and indexed in the Case Knowledge Base, (CKB);

- the statutes and regulations which are contained in the Statute Knowledge Base, (**SKB**), and the
- expert heuristic knowledge needed to reason with both the precedents and legislation. This type of knowledge is represented in two places, as inferential knowledge contained in rule sets in the Case Rule Base, (CRB), and as messages contained in the meta-slots of various object/classes.

Currently, the IKBALS II CKB contains thirty real legal cases with twenty dimensions that are the index to cases in the CKB. The SKB contains three rule sets with close to fifty rules per set. Most of the meta-knowledge used to drive the system is contained in the meta-slots of certain object/classes.

In IKBALS, which is a rule/object based system, problems are input in a form, based on what the Workcare experts have deemed 'important' attributes in determining Workcare disputes. Note, that the choice of these attributes is quite crucial to the success of the system. IKBALS II uses the rule/object base of IKBALS. When the current legal problem requiring a solution is input, a rule based engine determines which point-to-argue (ptas) apply to the current case. These ptas are then used to retrieve those cases that are indexed under the same ptas in the system. The pta's prerequisites determine what features to look for in a case. Since a case can be indexed under several different ptas, after a number of candidate resolved cases have been retrieved, the most similar case(s) is chosen.

Figures 2 and 3 illustrate the Case Knowledge Base Diagram. The case *Zuijs* vs. *Wirth Brothers Pty. Ltd.* encapsulates all the properties from the CaseObject hierarchy in addition to any extra properties found in the Employment Relationship Class.

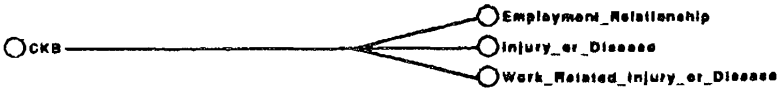


fig 2. Case Knowledge Base Diagram

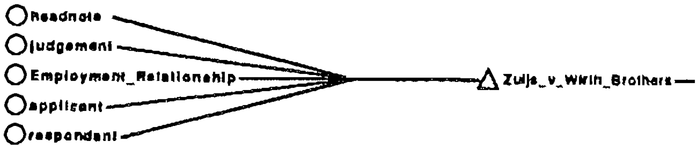


fig 3 Case Knowledge Base Diagram for *Zujs* vs. *Wirth Brothers*

The Statute Knowledge Base contains the normalised statutory rules used during the Rule Based inferencing process. Statutory rules are grouped into rule-sets depending on the type of relationship/concept they address. Currently, the rule-sets help elicitate three key factors needed to be proved before an entitlement to compensation can be deemed: namely the employment relationship, injury or disease and work related injury. For example, in the Employment Relationship rule-set, there is a collection of rules which deal with the statutory extension of the term 'worker'. This is required as the Accident Compensation Act contains provisions which extend the common law concept of employee, so that people who would otherwise be independent contractors will in certain circumstances be workers and therefore entitled to the benefits of the legislation. Figure 4. lists the categories of people to be tested under the term 'worker'.

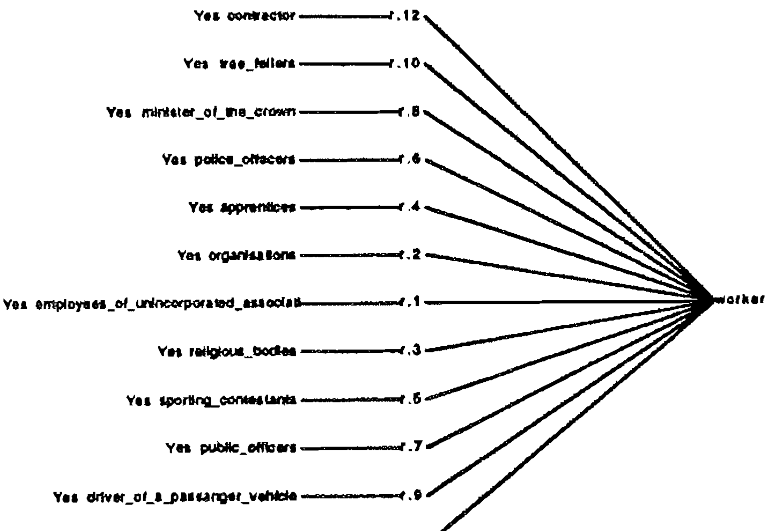


fig 4. Worker Diagram

4.3 Sample sessions using IKBALS

Because we are implementing legal expert systems, we believe it is useful to include some sample consultations. The first consultation [Vossos et. al,1991a], using IKBALSI advises on the likelihood of success for weekly payments.

The second consultation [Vossos et al., 1991b] uses IKBALS II, and compares the CIC to the thirty cases in our Case Knowledge Base. It then selects points to argue, and chooses the relevant precedents which a lawyer should consider in preparing his case. We shall only list those cases which IKBALSII deemed as relevant to the CIC. We shall also list an abstracted view of part of the CKB, and the "union meeting in course of employment" pta, both of which are relevant to our CIC.

In [*Johansen* v *ACC*] the worker was injured returning home form a strike meeting. The issues which arose were whether the worker's contract of employment was subsisting at the time of the injury, and if so, whether the injury arose out of that employment. Whilst the court decided that the worker contract was not terminated, it was also satisfied that the employer in no way condemned or gave permission either explicitly or implicitly to any person to strike or attend the strike meeting. Hence the Tribunal was not satisfied that there was a nexus between the worker's attendance at the meeting whilst on strike and her employment so as to establish that her attendance at the meeting was reasonably incidental to her employment.

In [*ACC* v *Gardiner*], the worker was injured while attending a union meeting which was not on the employer's premises and for which she was not being paid. Nevertheless, the Tribunal found that the injury occurred whilst the worker was performing an act incidental to the course of employment since she had the permission of the employer and indeed used the employer's vehicle to attend the meeting. The meeting was specifically called to discuss superannuation, an issue effecting employment conditions.

The case of [*Riego* v *ACC*] was decided on similar grounds. In this case, a worker was responding to a request made by the secretary of the union to come to work for the purposes of attending a union meeting. The worker's employer also requested the employee to attend the meeting. The meeting was scheduled for the worker's rostered day off, when the applicant was injured in a traffic accident on her way to work. The issue before the Tribunal was whether the injury which occurred on the way to the workplace was in the course of employment. The Tribunal found that her travel from her place of residence to the workplace was incidental to her employment obligations.

For each of the most relevant precedents, IKBALS II then proceeds to justify that the outcome of the current case should, (or shouldn't be) the same as the precedent's outcome. It does so by drawing the analogy between the two cases, focussing on their important similarities and differences.

In our [CIC] case, the following ptas were retrieved: action-incidental-to-employment, action-agreed-to-by-employer. Whilst the points meeting-not-on-employers-premises, and meeting-on-a-day-off are elements of the case, they are not ptas since if the employer agrees to his workers attending a union meeting during normal working hours then these points are irrelevant to the case.

The three cases IKBALSII chose were the above cases, primarily because they were the three that focused on the action-agreed-to-by-employer-pta i.e. the important issue appears to be whether the employer agreed (or not) to the employee taking the action that resulted in the injury.

5. CURRENT RESEARCH ISSUES FOR IKBALS II

In the previous sections, we have discussed our completed research involved in developing intelligent legal tools. The following projects are currently being undertaken.

5.1. Mediation and negotiation

Any intelligent legal advice requires the lawyer informing the client not only of the probability of winning the case, but also, the costs involved in doing so. Such costs could include time as well as money.

The IKBALS II system has a module that advises on costs involved in litigation under the act, and the time taken for the case to be heard. IKBALS II also offers advice on the amount the client is likely to receive. The client can compare this to the amount offered by the Accident Compensation Commission, and decide whether litigation is worthwhile.

An important issue for lawyers is that of alternative dispute resolution. Good lawyers look for alternatives to litigation. [Sycara, 1987] built the Persuader system, to offer mediation advice. Her system uses Case Based Reasoning, and facets of it are being incorporated in IKBALS.

5.2. Intelligent retrieval techniques

Much work is being performed on how to efficiently and intelligently retrieve cases. Because lawyers like user-friendly systems we are developing an interface to hypercard.

6. OTHER PROJECTS

In addition to the above research we are also implementing useful commercial rule-based systems. Such systems are proving useful for end-users, convince lawyers that Legal Expert Systems can be helpful, rather than a passing fad, and give our project further experience in rule based systems. These projects include:

. *Credit Act*

A system for determining if customers are eligible for financial support under the Credit Act.

. *Modelling instituting civil procedures*

A system that will help lawyers initiate litigation in civil disputes. These laws are rule-based, technical and often poorly understood by lawyers.

. *Eligibility for legal aid*

Currently legal aid lawyers spend much of their time determining if prospective clients are eligible for aid, rather than offering legal advice.

Our system advises on whether clients charged under the Crimes Act are eligible for Legal Aid. Since 59% of applicants for legal aid are charged under the Crimes Act, our system is very useful to the Legal Aid Commission. We are investigating building systems for the Family Law and Civil Litigation areas. These areas however involve fewer rules and more discretion, and will be more difficult to model.

. Comparing successful and unsuccessful negotiation - family law

As mentioned in 6.1., it is very important for lawyers to look at the prospects of mediation before they commence litigation. Most computer based mediation systems use Case Based Reasoning. Unfortunately successfully resolved cases are rarely recorded, since only contested cases are adjudicated, and hence recorded. The one exception is in the area of Family Law, whereupon dissolution of a marriage, all settlements whether contested or not, are filed with the court. We are constructing a system that will compare and contrast litigated and negotiated settlements. Our aim is to build a system that will allow us to quickly identify "problem cases", and offer sound mediation advice.

. Bail Act

We are currently modelling the Bail Act, with the aim of determining whether the act discriminates against any particular groups. The aim is to build a conceptual model of how magistrates reason when making determinations on bail, and then comparing this with the database of magistrates' decisions.

7. ACKNOWLEDGEMENTS

The driving force on this project has been George Vossos, who is currently completing a Phd. He has provided most of the technical and research advice in Computer Science. Valuable knowledge engineering expertise has been provided by Professor Tharam Dillon.

The project received important advice on the Workcare Act from Graeme Taylor, whilst Heather Aldred and Daniel Hunter provided the authors with sound advice about legal reasoning. Vivian Vossos and Michelle Nancarrow have been involved in translating the act into a machine readable form. Michael Aw, Chang Chen, Elizabeth May and Susie Tanevska have been involved in adding the extension modules to IKBALS II.

Don Berman and Kevin Ashley, gurus in the area, visited our laboratory in 1991. Don's 40 hour lecture series motivated a new generation of students to become devotees of this rapidly growing area.

Finally, thanks go to Alan Schwartz whose advice and encouragement led us to begin the project.

8. REFERENCES

Ashley, K. D. (1988), "Arguing by Analogy in Law: A Case Based Model", Analogical Reasoning, *Kluwer Academic Publishers*, pp 205-224.

Ashley, K.D., and Rissland, E.L. (1988), "A Case-Based Approach to Modelling Expertise", in *IEEE Expert*, Fall , pp 70-77.

Bain, W. M. (1986), "Case Based Reasoning: A Computer Model of Subjective Assessment", *PhD thesis,* Yale University.

Brodie, M.L. (1988), "Future Intelligent Information Systems: AI and Database Technologies working Together" in Brodie and Mylopolous (eds.) (1988) "Readings in Artificial Intelligence", 623-641.

Burton, S. J. (1985), "An Introduction to Law and Legal Reasoning", *Little, Brown and Company.*

Davis and Hamscher, "Model-Based Reasoning: Troubleshooting ", in "*Exploring Artificial Intelligence: Survey Talks from the National Conferences on Artificial Intelligence*", Morgan Kaufmann, 297-346.

Gardner, A. (1987) "An Artificial Intelligence Approach to Legal Reasoning", *Bradford/MIT Press* .

Hafner, C.D., Berman, D.H., (1990) "Artificial Intelligence and Law", *Encyclopedia of Computer Science and Technology,* Marcel Dekker Inc, Vol. 22, Supplement 77, pp 43-65.

Kolodner, J. (1988), "*Proceedings of a Workshop on Case Based Reasoning*", May 1988, Clearwater Beach, Florida.

Kowalski, R.A. (1979), "Logic for Problem Solving", *North Holland.*

Levi, E.H. (1948), "An Introduction to Legal Reasoning", *The University of Chicago Press.*

Moles, R. (1991) - *in this issue*

Riesbeck, C. K. and Schank, R. C. (1989) "*Inside Case Based Reasoning*", Lawrence Erlbaum Associates Publishers.

Rissland, E.L., Skalak, D.B. (1989) "Interpreting Statutory Predicates" *Proceedings of the Second International Conference on Artificial Intelligence and Law,* 1989, ACM Press, pp 46-53.

Schlobohm, D.A. and Waterman, D.A. (1987), "Explanation for an Expert System that Performs Estate Planning", *Proceedings First International Conference Artificial Intelligence and Law*, ACM Press, pp 18-27.

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 ACM,* Volume 29(5) , pp 370-386.

Skalak, D.B., (1989) "Taking Advantage of Models for Legal Classification", *Proceedings of the Second International*

Conference on Artificial Intelligence and Law, 1989, ACM Press, pp 234-241.

Slade, S. "Case-Based Reasoning: A Research Paradigm", *AI Magazine*, Vol. 12, No. 1, Spring 1991, pp 42-55.

Susskind,R.E. (1987) Expert Systems in Law, (A Jurisprudential Inquiry), *Clarendon Press*, Oxford.

Sycara, E. P., (1987), "Resolving Adversarial Conflicts: An Approach to Integrating Case *Based and Analytic Methods", Technical Report GIT-ICS-87/26*, Georgia Institute of Technology, School of Information and Computer Science, Atlanta, GA.

Thomas,D. (1989) "What's in an object", *BYTE*, March 1989.

Tyree, A. L., Greenleaf, G. and Mowbray, A., (1987), "Legal reasoning: the problem of precedent", *Proc. Conf. AJAI*, Sydney, November 1987, pp 419-432.

Vossos, G., Dillon, T., Zeleznikow, J. and Taylor, G. (1991a), "The Use of Object Oriented Principles to Develop Intelligent Legal Reasoning Systems", *Australian Computer Journal*, Vol. 23, No. 1, February 1991, pp 2-10.

Vossos, G., Zeleznikow, J., Dillon, T. and Vossos, V. (1991b), "An Example of Integrating Legal Case Based Reasoning with Object Oriented Rule-Based Systems - IKBALS II", *Proceedings of the Third International Conference on Artificial Intelligence and Law*, Oxford, June 25-28 1991, ACM Press, pp 91-101.

Waterman, D.A., Paul, J. and Peterson, M. (1986), "Expert Systems for Legal Decision Making", in *Applications of Expert Systems*, The Rand Corporation .

Yuen, H.S., Zeleznikow, J. and Dillon, T.S., "A Qualitative Data Model for Managing Knowledge in Organizational Planning"", *Proceedings of IMACS Workshop on Decision Support Systems and Qualitative Reasoning*, LAAS, Toulouse, France, March 1991, pp. 163-173.

BIOGRAPHY: John Zeleznikow is the Director of the Database Research Laboratory at the Applied Computing Research Institute, La Trobe University. He leads teams on developing intelligent information systems with special regard to law, public administration, health care and travel. He has a Phd in pure mathematics from Monash University, and has taught at numerous North American and French universities. He is an associate editor of the Australian Computer Journal, and has been chairman of many computer science conferences. He has over 30 research publications. He has a long interest in legal and political issues. He is also a sessional lecturer in the Faculty of Law at University of Melbourne.