Artificial Intelligence and Law **10:** 237–260, 2002.

An Australian perspective on research and development required for the construction of applied legal decision support systems

JOHN ZELEZNIKOW
Joseph Bell Centre for Forensic Statistics and Legal Reasoning, Faculty of Law, University of Edinburgh, Old College, South Bridge, Edinburgh, EH8 9YL, UK
E-mail: john.zeleznikow@ed.ac.uk

Abstract. At the Donald Berman Laboratory for Information Technology and Law, La Trobe University Australia,[1] we have been building legal decision support systems for a dozen years. Whilst most of our energy has been devoted to conducting research in Artificial Intelligence and Law, over the past few years we have increasingly focused upon building legal decision support systems that have a commercial focus. In this paper we discuss the evolution of our systems. We begin with a discussion of rule-based systems and discuss the transition to hybrid rule-based/case-based systems. We next discuss how we have used machine learning in building legal decision support systems. Our focus on using machine learning led us to investigate the domains of explanation and argumentation. We conclude by discussing our current work on building negotiation support systems and tools for constructing web-based legal decision support systems.

Key words: case-based reasoning, legal decision support systems, machine learning, negotiation, rule-based reasoning

1. Introduction

Over the past twelve years, we have focused upon research involved in the construction of decision support systems for legal professionals. Our goal has always been to investigate novel techniques for the construction of such systems.

Recently, there has been a marked shift in the way that governments fund academic research. Rather than only looking at the academic quality of research, governments have encouraged academics to find industrial partners. They have provided funds to encourage industry to work with universities.

The Australian Government, through a variety of programs, has moved research money from traditional to applied research. At the Donald Berman Laboratory for Information Technology and Law, we have taken advantage of the government's changing policies, to develop industry partnerships focussed upon the construction of legal decision support systems that have practical applications.

Our initial foray into the development of intelligent legal decision support systems, commenced with an examination of Sergot et al. (1986) which tried to automate the British Nationality Act of 1981 as a logic program. An examination of the British Nationality Act led us to realise that ideally, legal decision support systems must deal with the issue of open texture. Open texture was a concept first introduced by Waismann (1951) to assert that empirical concepts are necessarily indeterminate.

The concept of open texture is apt in the legal domain because new uses for terms, and new situations constantly arise in legal cases. Thus, as Berman and Hafner (1988) indicate, legal reasoning is essentially indeterminate because it is open-textured. Bench-Capon and Sergot (1988) view the indeterminacy in law as a specific consequence of the prevalence of open-textured terms. They define an open textured term as one whose extension or use cannot be determined in advance of its application.

Legal decision support systems need to apply to fields of law that range from characteristically discretionary on the one hand to rigid and rule like on the other. Furthermore, they must provide explanation and argumentation facilities.

To better understand the nature of legal decision support, open-texture and discretion, we have used a number of inferencing techniques: association rules, case-based reasoning, machine learning, neural networks and rule induction. Domains investigated include: Workers Compensation, Credit Law, Family Law Property Distribution, Family Law Mediation, Refugee Law, Eligibility for Legal Aid, Copyright Law, Eye-Witness Identification, Examining the causes of death (natural causes, suicide or homicide), Sentencing and the Building Industry.

Zeleznikow (2002b) states that one of the principal goals of the law is to reduce risk through the avoidance of litigation. Their motivation for building decision support systems that advise upon arbitration and mediation is to reduce the inherent risks involved in litigation. Zeleznikow (2002a) discusses how the provision of web-based legal decision support systems can help improve access to justice. We believe this trend is likely to facilitate the resolution of disputes outside litigation.

In this paper we discuss the research and development involved in building our suite of applicable legal decision support systems. Much of our work has been achieved by establishing a close collaboration with legal firms including Victoria Legal Aid (VLA) (Zeleznikow and Stranieri 2001). VLA based in Victoria, Australia is a government-funded provider of legal services for disadvantaged clients (http://www.legalaid.vic.gov.au). Its goals include providing legal aid in the most effective, economic and efficient manner and pursuing innovative means of providing legal services in the community.[2]

This paper is organised both chronologically and thematically. The reader can see how our approaches for building intelligent legal decision support systems have moved from constructing Logic Rule Based Systems to Object Oriented Systems and then to Case Based Systems. We then used both Machine Learning and Knowledge Discovery from Databases to understand how judges exercise discretion. Our

Table I. Legal decision support systems built at the Donald Berman Laboratory for Information Technology and Law during the last decade

System	Application	Reasoning techniques used	Status
IKBALS I	Workers Compensation	Rule-based reasoning and case-based reasoning	Abandoned due to change of legislation in 1992
CAAS	Credit Law	Rule-based reasoner	Used in back office of the Bank of Melbourne until 1995
IKBALS III	Credit Law	Rule-based reasoning and case-based reasoning. Rule induction was used to learn factors about closest cases	Solely a research prototype
Split Up	Family Law Property Distribution	Rule-based reasoning and neural networks. Separate argumentation shell developed	Prototype first version used privately by mediators, judges lawyers. Web-based second version is being developed for use by VLA
Family_Negotiator	Family Law Negotiation	Rule-based reasoning and case-based reasoning	Used solely to understand the domain of family mediation
Embrace	Refugee Law	Rule-based reasoning and information retrieval	Policy changes by a new government have meant the system is only used as a training tool
GetAid	Eligibility for Legal Aid	Uses sequenced transition networks and Argument Developer. Is placed on World Wide Web	The commercial success story!!! Is being used by VLA to provide on-line advice re eligibility for Legal Aid
RightCopy	Informs software developers of their copyright entitlements	Uses sequenced transition networks and Argument Developer.	SEA, our industry partner has chosen not to commercialise the system.
Sentencing Information System	Provides advice to VLA lawyers on possible sentences for criminals	Uses sequenced transition networks and Argument Developer	Under current development
Family_Winner	Family Law Negotiation	Rule-based reasoning, case-based reasoning and fuzzy cognitive maps	Under current development
ADVOKATE	Eye Witness Identification	Rule-based. Placed on WWW using JustReason	Under current development

current work involves using argumentation and constructing web-based systems, as well as developing negotiation support tools and performing on-line dispute resolution.

Table I illustrates decision support systems developed at the Donald Berman Laboratory for Information Technology and Law during the last decade.

2. Rule-based and hybrid intelligent legal decision support systems

The majority of commercially available legal decision support systems model fields of law that are complex but not discretionary. For example, SoftLaw, an Australian software house that specialises in legal knowledge based systems for the Australian Government has developed systems with tens of thousands of rules. It primarily builds systems with regard to entitlements, in domains such as social security legislation. There seems little doubt that the trend toward rule based systems to encode large and complex legislation will continue to a substantial extent, as claimed by SoftLaw (2000). This is assured by the increasing public demand for more transparency and consistency in government decision making alongside with the continuing enactment of increasingly complex legislation.

The German Conceptualist movement assumes that judges are almost totally constrained by rules. Every attempt is made by adherents to this theory to determine one single correct meaning for every term in every rule in a legal system. Once this is achieved, legal reasoning reduces to the logical application of facts to rules. However, few legal academics or professionals accept this view of law today.

Ironically, although many commentators including Moles and Dayal (1992) clearly express limitations of this approach for the majority of fields of law, rule based reasoning is still the predominant basis for legal decision support systems. The fundamental limitation not addressed by this view of law can be reduced to two significant omissions; the failure to model open texture and the failure to provide an analysis of how justification differs from the process used to arrive at decisions.

In constructing legal decision support systems, we have observed that most commercially successful systems have employed rules. The major reasons for this occurrence include that it is easy to model rules and the are many tools for building rule-based systems.

Given our desire to move beyond rule-based systems when modeling law, we commenced the IKBALS (*I*ntelligent *K*nowledge *Ba*sed *L*egal *S*ystems) project. IKBALS (Zeleznikow 1991) used the object-oriented approach to build a hybrid rule-based/case-based system to advise upon open texture in the domain of Workers Compensation. IKBALSI and IKBALSII both deal with statutory interpretation of the *Accident Compensation (General Amendment)* Act 1989 (Vic). The Act allows a worker who has been injured during employment to gain compensation for injuries suffered. These compensation payments are called WorkCare entitlements, and IKBALS focuses on elements giving rise to an entitlement.

The original prototype IKBALSI was a hybrid/object oriented rule-based system. Its descendant, IKBALSII, added case-based reasoning and intelligent information retrieval to the rule-based reasoner, through the use of a blackboard architecture.

The defeat of the Victorian Labour Government in October 1992 led to significant changes in the relevant legislation and abandonment of the specific system dealing with Workers' Compensation. However, we were still determined to use a hybrid agent architecture to build a legal knowledge based system and thus searched for suitable application areas and domain experts.

We were fortunate to find an interested legal partner in the Credit Law domain (Allan Moore of Allan Moore & Co). The resulting integrated deductive and analogical system system, IKBALSIII (Zeleznikow et al. 1994):

1. Includes induction as the basis for its case based retrieval function; and
2. Relies on distributed artificial intelligence techniques and the object oriented paradigm, rather than a blackboard architecture.

Induction is used in IKBALSIII to generate the indices into the cases. Thus, the developer can specify a number of cases, including the relevant factors and the outcome, and the induction algorithm will generate the indices automatically. This is an advance over other systems that require the developer to specify the indices manually.

Whilst simultaneously developing IKBALSIII, Vossos et al. (1993) developed the Credit Act Advisory System (CAAS). CAAS is a strictly rule based legal expert system which advises in relation to a small part of the Credit Act 1984 (Vic). CAAS does not argue directly with either statutes or precedents. Instead, it is a production rule system where the production rules forming the knowledge base of CAAS are heuristic rules supplied by experts from Allan Moore & Co. It was prototyped using NExpert Object and then compiled into C++ under Windows 3.1. In developing CAAS, Vossos et al. (1993) developed a framework for building commercial legal expert systems using C++, rather than the more expensive expert system shells. The system was commercially marketed to organisations involved in the provision of credit. Back-office employees at the Bank of Melbourne commercially used the system. CAAS provided advice as to whether a transaction is regulated, not regulated or exempt under the Act.

CAAS was essentially the rule-based component of IKBALSIII. Whilst CAAS, the rule-based part of IKBALS II covered the whole domain of the Victorian Credit Act (and was commercially exploited), IKBALS II only dealt with one possible open-textured predicate – *was the transaction for a business purpose*. IKBALSIII's novel technique for resolution of open texture had little commercial benefit.

3. Using knowledge discovery from databases to build intelligent legal decision support systems

Most successful Legal Decision Support Systems (for example systems built by Softlaw and JNANA) have primarily used rules. Rule-based systems are particularly useful for advising upon the allocation of benefits and entitlements (such as social security benefits, pension plans and child support allowance).

Of course, not all systems can be modeled using rules. Stranieri et al. (1999) have developed a jurisprudential theory focusing upon open texture and boundedness, which investigated those legal domains that can be modeled using rules.

3.1. SPLIT-UP

Split-Up provides advice on property distribution following divorce (Zeleznikow and Stranieri 1995). The aim of the approach used in developing Split-Up was to identify, with domain experts, relevant factors in the distribution of property under Australian family law. We then wanted to assemble a dataset of values on these factors from past cases that can be fed to machine learning programs such as neural networks. In this way, the manner that judges weighed factors in past cases

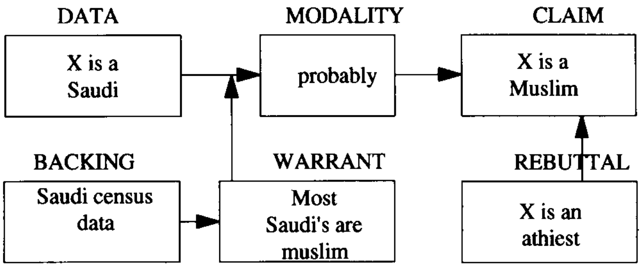

Figure 1. Toulmin argument structure.

could be learnt without the need to advance rules. The legal realist jurisprudence movement inspired this approach.

For legal realists exemplified by Llewellyn (1962), rules and principles may be invoked after a decision has been reached in order to ensure that a decision is just, moral and legally correct. Rules and principles are invoked to explain a decision but there is no need to assume they are used to reach the decision.

Ninety-four variables were identified as relevant for a determination in consultation with experts. The way the factors combine was not elicited from experts as rules or complex formulas. Rather, values on the 94 variables were to be extracted from cases previously decided, so that a neural network could learn to mimic the way in which judges had combined variables.

However, according to neural network rules of thumb, the number of cases needed to identify useful patterns given 94 relevant variables is in the many tens of thousands. Data from this number of cases is rarely available in any legal domain. Furthermore, few cases involve all 94 variables. For example, childless marriages have no values for all variables associated with children so a training set would be replete with missing values. In addition to this, it became obvious that the 94 variables were in no way independent.

In the Split-Up system, the relevant variables were structured as separate arguments following the argument structure advanced by Toulmin (1958). Toulmin concluded that all arguments, regardless of the domain, have a structure that consists of six basic invariants: claim, data, modality, rebuttal, warrant and backing. Every argument makes an assertion based on some data. The assertion of an argument stands as the claim of the argument. Knowing the data and the claim does not necessarily convince us that the claim follows from the data. A mechanism is required to act as a justification for the claim. This justification is known as the warrant. The backing supports the warrant and in a legal argument is typically a reference to a statute or a precedent case. The rebuttal component specifies an exception or condition that obviates the claim.

A survey of applications of the Toulmin Structure has revealed that the majority of researchers does not apply the original structure but vary it in one way or another. Figure 2 illustrates the structure representing the variation used in Split-Up. The rationale for the variations applied are described in Stranieri et al. (2001).

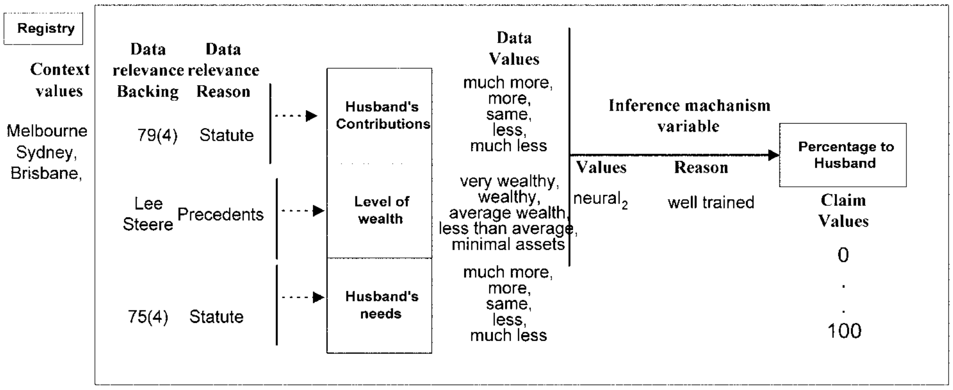

Figure 2. Generic argument for percentage split of assets to the husband.

Figure 2 illustrates one argument from the Split-Up system. We see from that figure that there are three data items. Each of these is the claim item of other arguments leading to a tree of arguments where the ultimate claim of the system is the root of the tree.

A key difference in our variation from the original is the specification of an inference mechanism variable. In the argument in Figure 2, the inference mechanism is a neural network. The network, once trained with appropriate past cases, will output a claim value (% split of assets) given values of the three data items.

In twenty of the thirty-five arguments in Split Up, claim values were inferred from data items with the use of neural networks whereas heuristics were used to infer claim values in the remaining arguments. The neural networks were trained from data from only 103 commonplace cases. This was possible because each argument involved a small number of data items due to the argument-based decomposition.

The Split-Up system produces an inference by the invocation of inference mechanisms stored in each argument. However, an explanation for an inference is generated after the event, in legal realist traditions by first invoking the data items that led to the claim. Additional explanatory text is supplied by reasons for relevance and backings. If the user questions either data item value, she is taken to the argument that generated that value as its claim.

The Split-Up system performed favorably on evaluation, despite the small number of samples. Because the law is constantly changing, it is important to update legal decision support systems. The original hybrid rule-based/neural network version of Split-Up was constructed in 1996. Currently, the tree of arguments is being modified in conjunction with domain experts from Victoria Legal Aid to accommodate recent changes in legislation. In particular

(a) The recent tendency by Family Court judges to view domestic violence as a negative financial contribution to a marriage.

(b) The re-introduction of spousal maintenance as a benefit to one of the partners. Under the *clean-break philosophy*, Family Court judges were reluctant to award spousal maintenance, since it would mean one partner would continue to be financially dependant on his/her ex-partner. However the increasing number of short, asset-poor, income-rich marriages has led to a re-consideration of the issue of spousal maintenance.
(c) The need to consider superannuation and pensions separately from other marital property.

The argument-based representation facilitates the localization of changes and makes maintenance feasible. The use of the argument based representation of knowledge enables machine learning techniques to be applied to model a field of law widely regarded as discretionary. The legal realist jurisprudence provided a justification for the separation of explanation from inference.

With the provision of domain expertise and financial support from VLA, we are currently developing a web-based version of Split-Up using the web-based shell ArgShell and the knowledge management tool JustReason. As a web-based system Split-Up will inform divorcees of their rights and support them to commence negotiations pertaining to their divorce.

3.2. RIGHTCOPY

The Toulmin Argument based structure initially developed in the Split-Up system and discussed in Stranieri et al. (2001) was used to develop RightCopy, an advisory system on computer copyright. The regulation of copyright in e-commerce presents significant legal, technological and social challenges. Commentators on the future of copyright law in cyber space disagree on the extent to which copyright law can remain appropriate in a digital environment. Some authors (e.g. Stallman 1994) advocate an overhaul of existing copyright principles. Others (e.g. Dixon and Self 1994) claim that very little change is needed at all. Existing copyright principles are, by and large, adequate for the digital age though minor adjustments are required. Although most governments are implementing minimal change (Stefik 1994), Richter and Chicola (1999) express concerns that the public interest aspects of copyright laws, typically implemented as fair use exceptions to authors rights are being eroded. Works locked with public key encryption are totally inaccessible by unauthorized users unless the key is cracked. This is so difficult that it remains virtually possible for an unauthorized user to gain access to a digital work even if the intended purpose underpinning the access is a legitimate fair use exception under copyright law.

Jurisprudence theories do not clearly provide insight for the development of practical systems in this area of law because it is changing so rapidly. To appreciate the changing nature of the law in the context of cyberspace we turned to the general concepts of regulation advanced by Lessig (1999). He claims that the regulation of any activity comes about due to law, social norms, economic forces or natural barriers. Lessig (1999) identifies four types of mechanisms that regulate social behaviour; the law, market forces, social norms and natural phenomena. A simple example illustrates these mechanisms. Motor vehicle speed along suburban streets may be regulated using the law by the implementation of an ordinance that sets a maximum speed, appropriate signage to in-form drivers, speed detection mechanisms and appropriate penalties for offenders. Motor vehicle speed may also be regulated with the installation of speed humps. In this case, no ordinance needs to be passed, no signage is needed and penalties are not relevant as the humps provide a natural barrier to speed. Similarly, the regulation of individual smoking could be realised by a legislative ban on smoking.

Natural barriers are proficient mechanisms for regulation. As Lessig (1999) points out there is no need for laws prohibiting the theft of skyscrapers because of the physical impossibility of stealing a building. In the physical world natural barriers such as humps on roads or un-moveable buildings are typically obvious. In cyberspace, natural barriers are implemented by software and are not so obvious. Lessig (1999) uses an example about chat rooms organised by a large, global internet service provider. The number of participants in a chat room is regulated by software that admits users up to the maximum number and displays a message inviting others to try later. The regulation of participants in chat rooms using software-restricting access is not as transparent as it would be if the regulation was implemented with laws, market incentives or social norms.

The formulation by Lessig (1999) motivated our search for a technological device that could helps to ensure fair use principles. Stranieri and Zeleznikow (2001) propose an agent based knowledge based approach to help regulate copyright. Five knowledge-based systems are described that are sufficiently flexible to protect authors rights without denying the public access to works for fair use purposes. The architecture involves the use of an agent-oriented approach. The owner of a work and users who wish to copy a portion of the work are participants in the discursive community and share the same generic arguments. In order to copy the work users constructs their own actual arguments. The agent representing the owner determines whether to release the work or not by constructing its own actual argument. The generic/actual framework simplifies the negotiation protocol and assists in the deployment of an agent oriented approach.

3.3. EMBRACE

Each year, vast numbers of individuals lodge applications to remain in Australia for fear of persecution if forced to return to their country of origin. Their claims are assessed on the basis of the United Nations Convention Relating to the Status of Refugees Refugee law is highly discretionary and extremely difficult to model. The United Nations Convention lists factors that are to be taken into account in reaching a determination but do not specify the weighting factors should have. For example, the convention recognises that an applicant must have a well founded fear of persecution on grounds of political opinion, race, religious beliefs, social group membership, or nationality but is silent on the interpretation or the relative weighting of these factors.

The Refugee Review Tribunal of Australia determines the refugee status of applicants. Ensuring that the decision making is as consistent as possible in this complex and discretionary domain is critical. Toward that end, Yearwood and Stranieri (1999) have modeled reasoning in refugee law as 200 generic arguments that have been used by members of the Refugee Review Tribunal or applicants for refugee status in recent years by perusing determinations and interviewing members. No inference mechanism has been specified for any generic arguments as no machine inferences can be entertained in this politically sensitive domain. However, even without machine inferences, the argument structures have proven to be useful in modeling refugee decisions. Furthermore, they have shown that the process of automatically generating a document that represents the reasoning made toward a decision is facilitated if the knowledge is represented as a series of actual arguments instantiated from generic templates.

Refugee Review Tribunal determinations are documents that express the reasoning steps a member of the Tribunal followed in order to infer conclusions regarding the status of an applicant. Although, it is reasonable to expect that a mapping between the reasoning steps used by a decision maker and the structure of the document produced would clearly be apparent, a number of authors have discovered that such a mapping is by no means obvious. In order to develop legal knowledge based systems that generate documents from their own reasoning steps, discourse analysis is invoked to bridge the gap and perform the mapping. Yearwood and Stranieri (1999) have identified a simple heuristic for traversing a series of actual arguments that often leads to a plausible document structure without the use of discourse analysis.

The resulting system, EMBRACE, supports a Refugee Review Tribunal member to make structured decisions that consider all relevant matter and only relevant material. This reduces the probability of an appeal against her decision.

4. Developing web-based legal decision support systems

We believe the development of on-line legal decision support systems has led to:

(i) Consistency – by replicating the manner in which decisions are made, decision support systems are encouraging the spreading of consistency in legal decision-making.

(ii) Transparency – by demonstrating how legal decisions are made, legal decision support systems are leading to a better community understanding of legal domains. This has the desired benefit of decreasing the level of public criticism of judicial decision making.

(iii) Efficiency – One of the major benefits of decision support systems is to make firms more efficient.

(iv) Enhanced support for dispute resolution – Users of legal decision support systems are aware of the likely outcome of litigation and thus are encouraged to avoid the costs and emotional stress of legal proceedings.

Whilst we do not claim that the construction of legal decision support systems will have a drastic effect on improving access to justice, we make the argument that the construction of such systems for community legal centres will improve their efficiency and increase the volume of advice they can offer. Until recently, most legal decision supports systems were rule-based and developed to run on personal computers. Whilst personal computer based tools are fine for lawyers, they may not be easily accessible to pro-se litigants. Reasons for this difficulty include their lack of an awareness of such systems, and the high cost of purchasing relevant software. Currently, very few legal decision support systems are available on the World Wide Web.

4.1. GETAID

When an applicant approaches VLA, his/her application is assessed to determine whether he/she should receive legal aid. This task chews up 60% of VLA's operating budget, yet provides no services to its clients. After passing a financial test, applicants for legal aid must pass a merit test. The merit test involves a prediction about the likely outcome of the case if it were to be decided by a Court. VLA grants officers, who have extensive experience in the practices of Victorian Courts, assess the merit test. This assessment involves the integration of procedural knowledge found in regulatory guidelines with expert lawyer knowledge that involves a considerable degree of discretion.

Figure 3 depicts a decision tree that represents reasoning used by VLA lawyers, to determine whether an applicant for legal aid, who is scheduled to appear in a minor (Magistrates) court, has met statutory guidelines.

Since experts could not readily represent knowledge about an applicant's prospects for acquittal as a decision tree, we decided to model the process as a tree of Toulmin arguments. The first of these is illustrated in Figure 4. In this figure only claim variables/values and data variable/values are included. During knowledge acquisition, the expert is prompted to articulate factors (data items) that may be relevant in determining a prospect for an acquittal claim, without any consideration about how the factors may combine to actually infer a claim value. For every factor presented, a reason for the item's relevance must be given. The next step in the knowledge acquisition exercise using the generic argument is to expand each data item. For example, the expert is asked to describe relevant factors for determining the strength of the crown case.

Once the tree is developed as far back as the expert regards appropriate for the task at hand, attention is then focussed on identifying one or more inference mechanisms that may be used to infer a claim value from data item values. It was difficult for the principal domain expert to articulate the ultimate argument (the

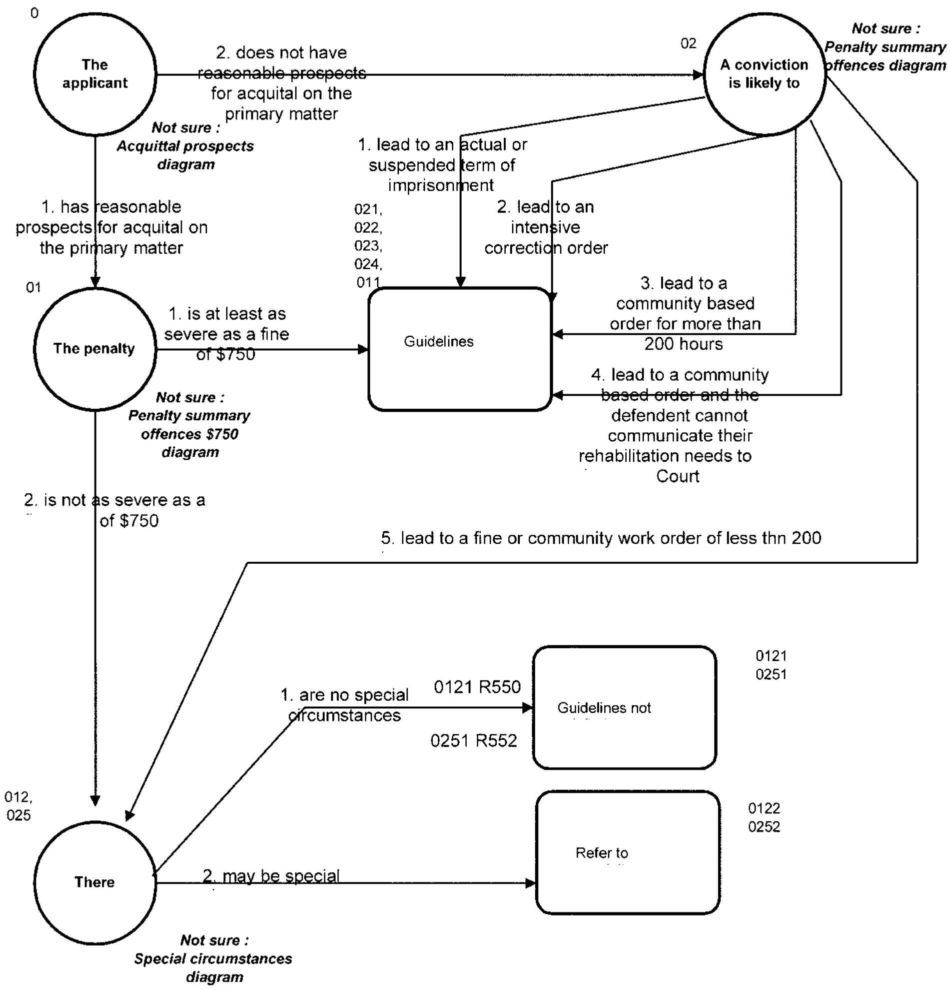

Figure 3. STN for eligibility for legal aid.

argument on the extreme right of Figure 4). She could not express her heuristic as rules because the way in which the factors combine is rarely made explicit. Her expertise was primarily a result of the experience she had gained in the domain. Although it is feasible to attempt to derive heuristics, the approach we used was to present a panel of experts with an exhaustive list of all combinations of data items as hypothetical cases and prompt the panel for a decision on acquittal prospects. Six experts and the knowledge engineer were able to record their decision in all of the exhaustive hypothetical cases (for that argument) in approximately 40 minutes. The decisions from each rater were merged to form a dataset of 600 records that were used to train neural networks.

The inference mechanism in WebShell consists of two components: a lookup table for exceptions and a weighted sum formula. Once the user has supplied values

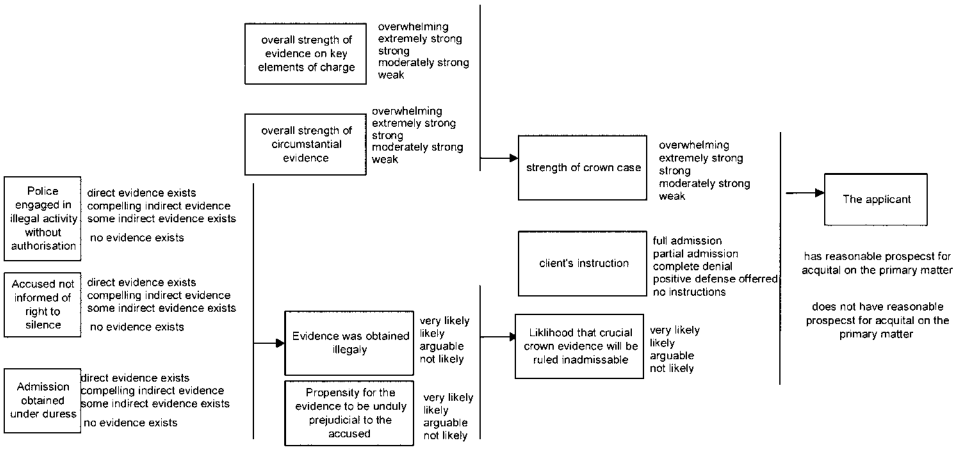

Figure 4. Argument tree for acquittal prospects.

for data items, the WebShell inference engine attempts to look up a claim value in the lookup table of exceptions. This table stores values that are exceptions to the weighted sum formula that are detected during the evaluation phase of knowledge based system development. If no entry is found in the lookup table, the inference engine applies a weighted sum formula according to weights associated with each data item. Using a lookup table to store the mapping between data values and claim values also enables the use of inference methods other than neural networks.

Neural network inferences can be implemented by storing all possible data item inputs and corresponding claim value outputs in the lookup table. A real time, web-based implementation cannot rebuild a neural network for each inference without causing consultation delays so storing all inputs/outputs as a lookup table enables fast inferences even when the source was a neural network.

A user consults the GetAid system via the web pages that are generated from the decision tree described above. Suppose a user follows the ‘Not-Sure’ link on the web page depicted in Figure 4. She is taken to a page that presents three user prompts that derive directly from the argument depicted in Figure 4; *strength of the crown case*, *client’s instruction and likelihood that crown evidence is ruled inadmissible*. This page is illustrated in Figure 5. The user is presented with a consistent user interface throughout and is generally unaware that some pages are generated from the argument tree and others from the decision tree.

The PHP program that implements the argument based inferences is somewhat more complex than the STN but it is still a small and relatively simple program that executes on the server side very quickly and is not memory intensive. The GetAid was tested by VLA experts and developed in conjunction with web-based lodgement of applications for legal aid (Hall et al. 2002). Commencing the middle of 2003, VLA clients will use the GetAid system.

Figure 5. WebShell based on Argument tree.

Figure 6. WebShell based on Argument tree.

4.2. ADVOKATE

Having developed a suite of legal knowledge based systems, we noted that there are certain generic principles involved in constructing such systems. To support the development of our systems, we have established an Australian, not for profit start up company JUSTSYS (www.justsys.com.au). JUSTSYS developed the GetAid prototype into a working system using Webshell and ArgumentDeveloper and is currently using these tools to build other web-based legal decision support systems. One significant example is the development of a web-based sentencing information system to support VLA barristers make arguments with regard to sentencing decisions. Most other Sentencing Information Systems have focussed upon providing decision support for judges (Zeleznikow 2000).

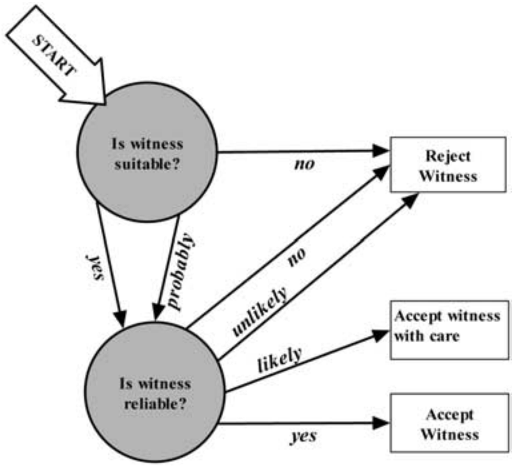

Figure 7. ADVOKATE: Rule-based section of the knowledge model.

ADVOKATE is a web-enabled knowledge-based decision support application designed for use in criminal investigations, civil litigation or as a teaching aide for investigative training. ADVOKATE provides an indicative assessment of the credibility of eyewitness testimony. The model was formatively and rapidly evaluated by using it as the basis for a prototype web deployed application built using the expert system shell, WebShell. The knowledge model supporting ADVOKATE was developed using knowledge acquired from various sources including legal cases, interviews with domain experts and reports in the literature. Empirical research, such as the amount of time the witness observed the perpetrator, or the effects of time delay, was also incorporated in the model. The knowledge model is presented in two parts, the first dealing with rule-based inferences and the second with knowledge containing some elements of discretionary inferencing. Figure 7 demonstrates how directed graph techniques are used to model rule-based knowledge. This simple directed graph demonstrates that the credibility of an eyewitness involves a preliminary assessment of suitability followed by the principal analysis for reliability. Failure in either of the two tests suggests that the eyewitness is not credible and should be rejected.

The detailed determination of witness' suitability and reliability are not rule-based inferences but rather discretionary decisions where the decision maker, taking account of several input factors, chooses from one of several possible outcomes. Decision-makers may arrive at different outcomes, depending on how they choose to inference from their understanding of the input factors. Thus the ADVOKATE domain can be categorised as a bounded discretionary domain (Zeleznikow 2000). The factors to be taken into account are known but no norms specified, leaving the decision-maker free to weight the factors, as they so desire.

Decisions with some discretionary elements are modeled using Toulmin Arguments (Stranieri et al. 2001). Argument trees are used to further refine the

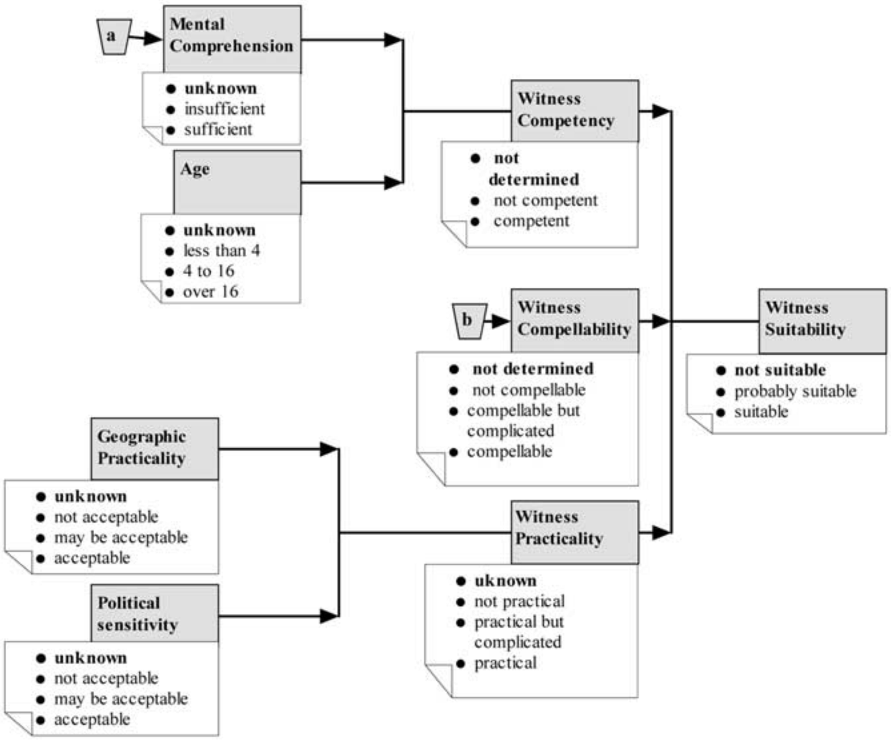

Figure 8. ADVOKATE: Part of the knowledge model of witness suitability.

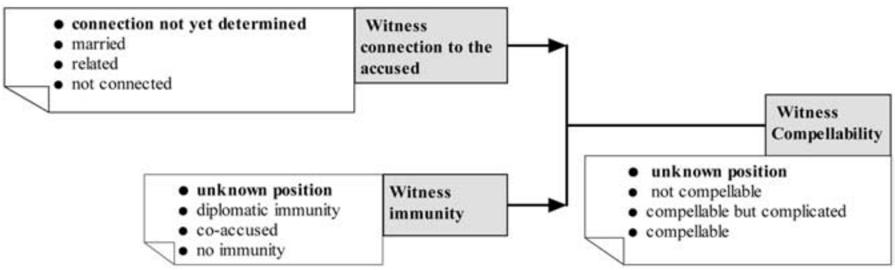

Figure 9. ADVOKATE: Witness Compellability.

knowledge depicted as directed graph nodes in Figure 7. Figures 8 and 9 show part of the argument tree for witness suitability and Figure 10 models witness reliability argumentation. The Toulmin warrant component is here replaced with inference mechanisms and a reason for relevance, however, how the factors are considered and combined by a decision maker when determining a claim, is not depicted in this model.

ADVOKATE, implemented as a browser accessible application, was made available to forensic experts, lawyers and police who provided feedback to the de-

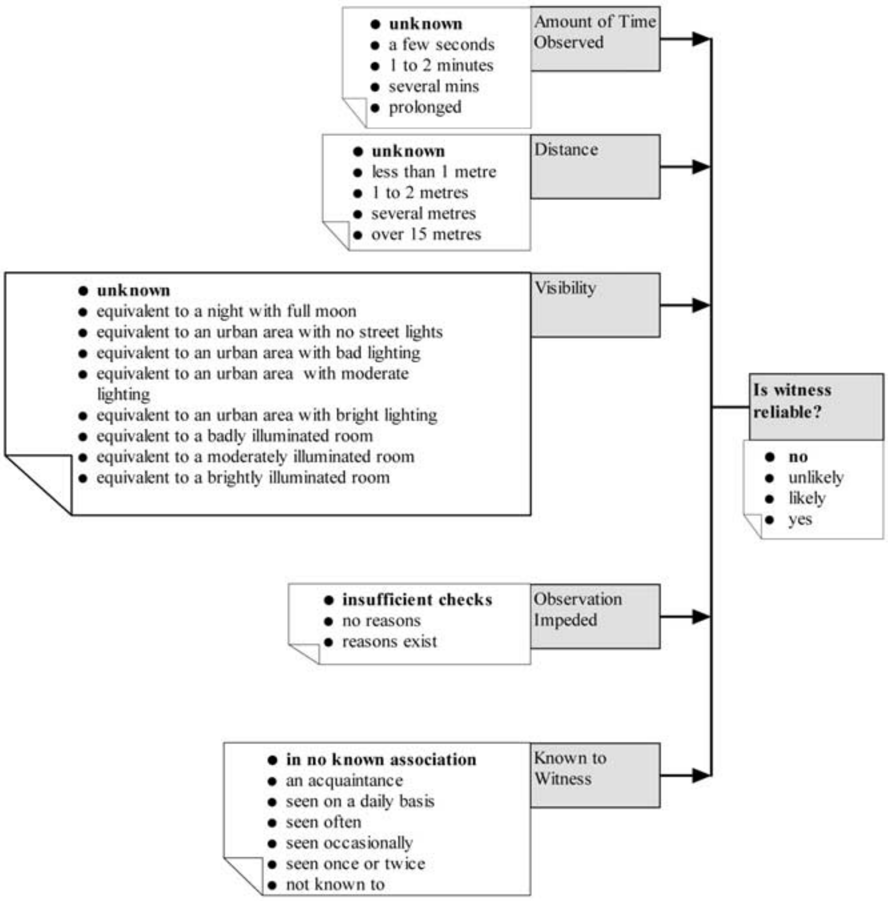

Figure 10. ADVOKATE: Witness Reliability knowledge model.

signers. The knowledge model was iteratively refined and enhanced and is available at http://advokate.bromby.vze.com/.

5. Negotiation support systems and on-line dispute resolution

Ross (1980) states the *principal institution of the law is not trial*; *it is settlement out of court*. Williams (1983) notes that whilst the figures may vary in different jurisdictions, of all the cases listed before the courts only about 5% of the cases are ever heard by the court and only 1% of the cases result in judicial decision-making. Thus it is important to develop Legal Decision Support Systems that support negotiation.

Katsh and Rifkin (2001) state that compared to litigation, Alternative Dispute Resolution has the following advantages:

(a) Lower cost;
(b) Greater speed;

(c) More flexibility in outcomes;
(d) Less adversarial;
(e) More informal;
(f) Solution rather than blame-oriented;
(g) Private.

To avoid the risks of extra costs and an unfavourable outcome, disputants often prefer to negotiate rather than litigate. Whilst investigating how disputants evaluate the risks of litigation researchers are faced with a basic hurdle – outcomes are often, indeed usually, kept secret. If the case is litigated, it could be used as a precedent for future cases, which may be a disincentive for one or more of the litigants. Publicity of cases and the norms resulting from cases makes the public aware of the changing attitudes towards legal issues. The adjudication decision not only leads to the resolution of the dispute between the parties, but it also provides norms for changing community values. This latter facet is lost in negotiated settlements.

The secrecy behind negotiated settlements is one of the reasons for the paucity of published material on legal decision support systems dealing with risk. JNANA (http://www.jnana.com) was founded in 1995 as Counselware, with the aim of building decision support systems for lawyers. The company very quickly realised that there was a large commercial need for decision support systems that advise upon risk assessment. Such systems are not made available to the public. JNANA currently focuses upon building a software platform to enable advice to be deployed over the Internet and Intranet. JNANA is now being used broadly in many industries, such as financial services, health care, customer relationship management, legal, and regulatory compliance.

We are building systems that support Alternative Dispute Resolution by advising on BATNAS, suggesting trade-offs and structuring compromises. Trade-off and compromise suggestions use point allocation and game theory. Fundamental to the concept of principled negotiation (Fisher and Ury 1981) is the notion of *Know your best alternative to a negotiated agreement (BATNA)*. The reason you negotiate with someone is to produce better results than would otherwise occur. If you are unaware of what results you could obtain if the negotiations are unsuccessful, you run the risk of: (1) Entering into an agreement that you would be better off rejecting; OR (2) Rejecting an agreement you would be better off entering into.

6. Split-Up as a negotiation support system

Split-Up can be used to determine one's BATNA for a negotiation. It first shows both litigants what they would be expected to be awarded by a court if their relative claims were accepted. It gives them relevant advice as to what would happen if some or all of their claims were rejected. They are able to have dialogues with the Split-Up system about hypothetical situations, which would support their negotiation. Both litigants then have clear ideas about the strengths and weakness of their claims.

Table II. Use of Split-Up to provide negotiation advice

	W's%	H's %
Given one accepts W's beliefs	65	35
Given one accepts H's beliefs	42	58
Given one accepts H's beliefs but gives W custody of the children	60	40

For example, suppose the disputants' goals are entered into the system to determine the asset distributions for both W and H in a hypothetical example (Bellucci and Zeleznikow 2001). The Split-Up system then provided the following answers as to the percentages of the marital assets received by each party:

Clearly custody of the children is very significant in determining the husband's property distribution. If he were unlikely to win custody of the children, the husband would be well advised to accept 40% of the common pool (otherwise he would also risk paying large legal fees and having on-going conflict).

Jennings et al. (2001) developed a generic framework for classifying and viewing automated negotiations. This framework was then used to analyse the three main methods of approach that have been adopted to automated negotiation, namely:

1. Game theory
2. Heuristics
3. Argumentation based approaches.

6.1. FAMILY_WINNER

Bellucci and Zeleznikow (2001) have used all three techniques in building negotiation support systems. They have focused upon Australian Family Law.

Split-Up uses an argumentation-based approach to advise disputants upon their BATNA.

Game theoretic techniques and decision theory were the basis for AdjustedWinner (Bellucci and Zeleznikow 1998), which implemented the procedure of Brams and Taylor (1996). AdjustedWinner is a point allocation procedure that distributes items or issues to people on the premise of whoever values the item or issue more. The two players are required to explicitly indicate how much they value each of the different issues by distributing 100 points across the range of issues in dispute. The Adjusted Winner paradigm is a fair and equitable procedure. At the end of allocation of assets, each party accrues the same number of points. It often leads to a win-win situation. Although the system suggests a suitable allocation of items or issues, it is up to the human negotiators to finalise the agreement acceptable to both parties.

Arising from our work on the AdjustedWinner algorithm, we have noted that

1. The more issues and sub-issues in dispute, the easier it is to form trade-offs and hence reach a negotiated agreement;

2. We choose as the first issue to resolve the issue on which the disputants are furthest apart – one wants it greatly, the other considerably less so.

Family_Winner (Bellucci and Zeleznikow 2001) uses both game theory and heuristics. It supports the process of negotiation by introducing importance values to indicate the degree to which each party desires to be awarded the issue being considered. The system uses this information to form trade-off rules. The trade-off rules are used to allocate issues according to the logrolling strategy. The system makes this analysis by transforming user input into trade-off values, used directly on trade-off maps, which show the effect of an issue's allocation on all unallocated issues.

Users of the Family_Winner system enter information such as the issues disputed, indications of their importance to the respective parties and how the issues relate to each other. An analysis of the aforementioned information is compiled, which is then translated into graphical trade-off maps. The maps illustrate the relevant issues, their importance to each party and trade-off capabilities of each issue. The system takes into account the dynamics of negotiation by representing the relations that exist between issues. Maps are developed by the system to show a negotiator's preferences and relation strengths between issues. It is from these maps that trade-offs and compromises can be enacted, resulting in changes to the initial values placed on issues.

The user is asked if the issues can be resolved in its current form. If so, the system then proceeds to allocate the issue as desired by the parties. Otherwise, the user is asked to decompose an issue chosen by the system as the least contentious. Essentially the issue on which there is the least disagreement (one party requires it greatly whilst the other party expresses little interest in the issue) is chosen to be the issue first considered. Users are asked to enter sub-issues. As issues are decomposed, they are stored in a decomposition hierarchy, with all links intact. This structure has been put in place to recognise there may be sub-issues within issues on which agreement can be attained. It is important to note that the greater the number of issues in dispute, the easier it may be to allocate issues, as the possibility of trade-offs increases. This may seem counter intuitive, but if only one issue needs to be resolved, then suggesting trade-offs is not possible.

This process of decomposition continues through the one issue, until the users decide the current level is the lowest decomposition possible. At this point, the system calculates which issue to allocate to which party, then removes this issue from the parties respective trade-off maps, and makes appropriate numerical adjustments to remaining issues linked to the issue just allocated. The resulting trade-off maps are displayed to the users, so they can see what trade-offs are made in the allocation of issues. When all issues are allocated at the one level, then decomposition of issues continues, re-commencing from the top level in a sequential manner.

The algorithms implemented in the system support the process of negotiation by introducing importance values to indicate the degree to which each party desires to be awarded each issue. It is assumed that the importance value of an issue is directly related to how much the disputant wants the issue to be awarded to her. The system uses this information to form trade-off rules. Systems such as Family_Winner are offer far more negotiation support than decision support systems that advise upon BATNAs.

We are building web-based systems that offer advice, rather than template-based systems such as Intersettle and Cybersettle.

6.2. BUILDING ADVISORY SYSTEM

We have used data mining (Stranieri and Zeleznikow 2003) to build numerous legal and negotiation decision support systems. In the Building Industry Advisor Project we are using data mining to build a web-based decision support system for use in improving the consistency and predictability of adjudicators' decisions in building construction disputes.

At Glasgow Caledonian University we have commenced a project on the Development and testing of a United Kingdom web-based decision support system for use in improving the consistency and predictability of adjudicators' decisions in building construction disputes. We are building a web-based decision support system by

(a) Combining the records of project partners (The Adjudication Reporting Centre; James R Knowles plc, Construction Contracts Consultants; MacRoberts, Solicitors; and Bishops, Solicitors); electronically publishing these records; creating a standard hub where stakeholders can record adjudication data; and data mining the records;
(b) decision modeling of the domain of building industry dispute resolution by developing a web-based model of legal reasoning in adjudication;
(c) commissioning a tool for predicting the course of building dispute adjudications.

The project has as objectives:

1. The data mining of adjudication records. The decisions of adjudicators is being examined for coherence. The major impact of this will be to replace the largely anecdotal experience of adjudication to a systematic and transparent analysis. Data is being collated into categories that serve not only for statistical analysis, but is also being collated from the point of view of the legal issues concerned and how these were dealt with by the adjudicator.
2. The development of new web-based, analytical tools to identify the predictability of decisions involving many variables, some of which will depend on legal reasoning, and some on purely statistical analysis. Accordingly, a decision model is being created based on a legalistic approach to adjudication, as derived from over one hundred decisions that have been made by the UK

courts. This will enable stakeholders to access the model to anticipate likely outcomes in an adjudication, and for adjudicators to test their decisions.
3. Testing the decision model against new adjudications which are being handled by project partners, and the identification of issues where the model aids predictability, and which may be used to facilitate settlement and reduce conflict.

The impact of the achieving of these objectives will be to have greater transparency as to the cause of disputes that go to adjudication and the likely result of such disputes. Predictability should serve diminish legal wrangling, with the ensuing saving of legal costs. The beneficiaries will be all stakeholders in the construction industry, but particularly smaller companies for whom adjudication was designed to help in the first place, thereby helping smaller companies to adopt business improvement measures. There are also teaching and self-help training applications, and the web site would provide a one-stop source of trend data on adjudications.

The project will also provide an easily accessed, one-stop decision support system with information on procedure and case law and is capable of being used by adjudicators to test reasoning before publishing a decision. The opportunities it will bring include: (1) Ability to test reasoning; (2) Ability to test likelihood of success; (3) Information and guidance on procedure, thereby reducing the danger of procedural error by adjudicators; (4) Up-to-date information and guidance on matters that frequently come to adjudication; (5) Up-to-date information and guidance on case law; (6) Training tool; (7) Reporter on trends.

7. Conclusion

In this paper we have presented a detailed discussion of legal knowledge based systems developed at the Donald Berman Laboratory for Information Technology and Law. Only a few of the systems have been commercialised. Reasons for a decision to produce commercial software are primarily based on the desire of the industry partner, rather than the performance of the software.

In developing a multitude of systems, in Workers Compensation, Credit Law, Family Law Property Distribution, Mediation and Negotiation, Refugee Law, Eligibility for Legal Aid, Copyright Law and Eye-Witness Identification we have noticed the need to build generic tools for building web-based systems. This is the focus of both our current research and development work being conducted at JUSTSYS.

Notes

[1] And associated organisations such the as School of Information Technology and Mathematical Sciences, at the University of Ballarat and the Joseph Bell Centre for Forensic Statistics and Legal reasoning at the University of Edinburgh.

[2] As set out in the Legal Aid Act 1978.

References

Bellucci, E. and Zeleznikow, J. (2001). Representations for Decision Making Support in Negotiation. Journal of Decision Support 10(3–4): 449–479.

Bench-Capon, T. J. M. and Sergot, M. J. (1988). Towards a Rule-Based Representation of Open Texture in Law. In Walter, C. (ed.) Computer Power and Legal Language, 39–61. Quorum Books: New York.

Berman, D. H. and Hafner, C. D. (1988). Obstacles to the Development of Logic-Based Models of Legal Reasoning. In Walter, C. (ed.) Computer Power and Legal Reasoning, 183–214. Quorum Books: New York.

Bromby, M. C. and Hall, M. J. J. (2002). The Development and Rapid Evaluation of the Knowledge Model of ADVOKATE: An Advisory System to Assess the Credibility of Eyewitness Testimony. In Proceedings of Eleventh International Conference on Legal Knowledge Based Systems, 143–152. IOS Publications: Amsterdam.

Dixon, A. and Self, L. (1994). Copyright Protection for the Information Superhighway. In Firth, A., Lane, S and Smythe, Y. (eds) 1998. Readings in Intellectual Property. A Selection of Articles from EIPR and Ent. L. R. Sweet and Maxwell: London.

Jennings, N. R., Faratin, P., Lomuscio, A.R., Parsons, S., Wooldridge, M. J., and Sierra, C. (2001). Automated Negotiation: Prospects, Methods and Challenges. Group Decision and Negotiation 10(2): 199–215.

Katsh, E. and Rifkin, J. (2001). Online Dispute Resolution: Resolving Conflicts in Cyberspace. Jossey-Bass: San Francisco.

Lessig, L. (1999). Code and Other Laws of Cyberspace. Basic Books.

Llewellyn, K. (1962). Jurisprudence. University of Chicago Press.

Moles, R. N. and Dayal, S. (1992). There Is More to Life than Logic. Journal of Law and Information Science 3(2): 188–218.

Richter, K., K., Chicola, J, M. (1999). Digital Rights for Intellectual Property Protection. In Proceedings of the IASTED International Conference. Law and Technology (LawTech'99), 27–31. ACTA Press.

Ross, H. L. (1980). Settled Out of Court. Aldine.

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: 370–386.

Softlaw (2000). http://www.softlaw.com.au. Accessed June 1, 2001.

Stallman R. (1994). Why Software Should Not Have Owners, at http://www.gnu.org/philosophy/why-free.html

Stefik, M. (1997). Shifting The Possible: How Trusted Systems and Digital Property Rights Challenge Us to Rethink Digital Publishing, Berkeley Technology Law Journal 12, 1 (Spring).

Stranieri, A. and Zeleznikow, J. (2001). Copyright Regulation with Argumentation Agents. Information and Communications Technology Law 10(1): 109–123.

Stranieri, A., Zeleznikow, J., Gawler, M., and Lewis, B. (1999). A Hybrid-Neural Approach to the Automation of Legal Reasoning in the Discretionary Domain of Family Law in Australia. Artificial Intelligence and Law 7(2–3): 153-183.

Stranieri, A., Zeleznikow, J., and Yearwood, J. (2001). Argumentation Structures that Integrate Dialectical and Monoletical Reasoning. Knowledge Engineering Review 16(4): 331–348.

Toulmin, S. (1958). The Uses of Argument. Cambridge University Press: Cambridge.

Vossos, G., Zeleznikow, J., Moore, A., and Hunter, D. (1993). The Credit Act Advisory System (CAAS): Conversion from an Expert System Prototype to a C++ Commercial System. In Proceedings of the Fourth International Conference on Artificial Intelligence and Law, 180–183. ACM Press: Amsterdam.

Waismann, F. (1951). Verifiability. In Flew, A (ed), Logic and Language. Blackwell.

Williams, G. R. (1983). Legal Negotiation and Settlement. West Publishing Co.: St. Paul, Minnesota.

Yearwood, J. and Stranieri, A. (1999). The Integration of Retrieval, Reasoning and Drafting for Refugee Law: A Third Generation Legal Knowledge Based System. In Proceedings of Seventh International Conference on Artificial Intelligence and Law, 117–126. ACM: Oslo.

Zeleznikow, J. (1991). Building Intelligent Legal Tools – The IKBALS Project. Journal of Law and Information Science 2(2): 165–184.

Zeleznikow, J. (2000). Building Judicial Decision Support Systems in Discretionary Legal Domains. International Review of Law, Computers and Technology 14(3): 341–356.

Zeleznikow, J. (2002a). Using Web-Based Legal Decision Support Systems to Improve Access to Justice. Information and Communications Technology Law 11(1): 15–33.

Zeleznikow, J. (2002b). Risk, Negotiation and Argumentation – A Decision Support System Based Approach. Law, Probability and Risk 1: 37–48.

Zeleznikow, J. and Stranieri, A. (1995). The Split-Up System: Integrating Neural Networks and Rule-Based Reasoning in the Legal Domain. In Proceedings of Fifth International Conference on Artificial Intelligence and Law, 185–194. ACM.

Zeleznikow, J. and Stranieri, A. (2001). The Use of Legal Decision Support Systems at Victoria Legal Aid. In Proceedings of ISDSS2001– Sixth International Conference on Decision Support Systems, 18-192. Brunel University: London.

Zeleznikow, J., Vossos, G., and Hunter, D. (1994). The IKBALS Project: Multimodal Reasoning in Legal Knowledge Based Systems. Artificial Intelligence and Law 2(3): 169–203.