Developing computational models of discretion to build legal knowledge based systems

Yaakov HaCohen Kerner, Uri Schild and John Zeleznikow[1]

Jerusalem College of Technology, 21 Havaad Haleumi St., P.O.B. 16031, 91160 Jerusalem; Department of Mathematics and Computer Science, Bar Ilan University, Ramat Gan, Israel, 52900; Department of Mathematics and Computer Science, Bar Ilan University, Ramat Gan, Israel, 52900

kerner@avoda.jct.ac.il; schild@cs.biu.ac.il; johnz@cs.biu.ac.il

Abstract

Few legal knowledge based systems have been constructed which provide numerical advice. None have been built in discretionary domains. Our research, directed towards the domains of sentencing and family law property division has lead to the development of three distinct forms of judicial discretion.

To model these different discretionary domains we use diverse artificial intelligence tools including case-based reasoning and knowledge discovery from databases. We carry out a detailed comparison of two discretionary legal knowledge based systems. Judge's Apprentice is a case-based reasoner which recommends ranges of sentences for convicted Israeli rapists and robbers. SplitUp uses Knowledge Discovery from Databases to learn what percentage of marital property the partners to a divorce in Australia will receive. The systems are compared with regard to reasoning, explanation, evaluation and coping with conflicting cases.

Keywords: Discretion, classifying discretionary domains, learning from cases, explanation, evaluation

1. Introduction

Discretion is a power or right conferred upon decision-makers to act according to the dictates of their own judgement and conscience, uncontrolled by the judgement or conscience of others. Nevertheless, decision-makers must in accordance with the rule of law and their decisions must preserve the rights of all parties effected by the decision-making. It is thus essential that discretionary decision-making not be arbitrary û since an arbitrary application of discretion could lead to enhanced conflict by any aggrieved parties. Rather than appear to make random decisions, they wish to develop a measure of consistency to their decision-making.

Four important areas of law in which magistrates and judges are given wide discretionary powers are the granting of bail, criminal sentencing, the granting of refugee status and most aspects of Family Law. Save for the work of Hassett we are not aware of significant support systems which provide advice about bail. Yearwood [20] has built a system to help retrieve prior cases in the domain of refugee law. Waterman and Peterson [19] developed LDS (Legal Decision Making system) which assisted legal experts in settling product liability costs. Given a description of a product liability case, LDS calculates defendant liability, case worth and an 'equitable' settlement case. Otherwise, there have been very few legal knowledge based systems which provide numerical computational advice.

To examine whether discretionary domains should in fact be modelled, [16] have complemented Hart's concept of open texturedness [3] with the notion of boundedness. They also introduced the distinction between landmark and commonplace cases, to enable them to perform knowledge discovery in open textured but bounded domains. Whilst we do not claim to have developed a computational model for judicial discretion making, our research has lead us to extend the work of [16] by developing the notion that there are three distinct forms of judicial discretion: narrow discretion, bounded discretion and unfettered discretion.

[1] Author's permanent address is Applied Computing Research Institute, La Trobe University, Bundoora, Victoria, Australia, 3083: johnz@latcs1.cs.latrobe.edu.au

We illustrate our theory of developing computational models for judicial discretion making, by considering two specific domains. Judge's Apprentice advises upon the sentencing of criminals found guilty of rape or robbery in Israel. SplitUp [16] performs the division of marital property following divorce in Australia. Both these systems reason from cases and provide numerical answers.

The development of legal support systems in discretionary domains will have numerous benefits for society. It will lead to: (i) enhanced consistency in decision-making; (ii) community understanding of the domain and hence less public criticism of judicial decision making; (iii) support for dispute resolution, since users of the system will be aware of the likely outcome of litigation and thus be encouraged to avoid the costs and emotional stress of legal proceedings.

2. Computational models of discretion

2.1 Discretion in Australian Family Law

In determining the distribution of property under the Family Law Act (1975) a judge performs the following functions:

1. She determines the assets of the marriage the Court is empowered to distribute. This task is known as common pool determination;

2. She determines what percentage of the common pool each party is empowered to receive;

3. She determines a final property order in line with the decisions made in 1 and 2.

A hypothetical panel of Family Court judges who agree on all facts of a particular divorce can conceivably arrive at different percentages of the assets that ought to be awarded to the wife. Different outcomes may be due to the presence of vague terms or defeasible rules. For example, one judge classifies a lottery win as a contribution to the marriage whereas another judge decides otherwise. A third judge applies the principle of an *asset by asset* approach whereas a fourth judge considers that principle irrelevant and adopts the *global* approach.[2] In discretionary domains it is possible that outcomes may differ even if there are no classification anomalies and all judges have used the same principles, because judges apply different weights to each relevant factor. The statute clearly affords the decision-maker precisely this sort of discretion.

The presence of discretion indicates that two or more legitimate judgements may have identical findings of fact yet different outcomes. Kovacs [6] notes that the principal statute underpinning family law, the Family Law Act (1975) presents a *'shopping list'* of factors to be taken into account in arriving at an order that distributes matrimonial property to divorcees. The relative importance of each factor remains unspecified and many crucial terms are not defined. For example, the age, state of health and financial resources of the litigants are explicitly mentioned in the statute as relevant factors yet their relative weightings are unspecified. The Act clearly allows the decision-maker a great deal of discretion in interpreting and weighing factors.

The notion of a shopping list of factors is crucial in the exercise of judicial discretion. Judges in the Australian Family Court and Israeli Criminal Courts reach their decisions by giving the factors weights. If a domain does not have a shopping list of factors, then we will claim it is impossible to model.

2.2 Sentencing in Israel

[Schild 1998] claims that in the domain of criminal sentencing even when giving more or less identical reasons, judges arrive at very different conclusions. In many instances there is a great disparity in the decisions of sentencing judges û even in quite similar cases. The same judge often decides on vastly different sentences in similar cases even when they occur over a relatively short time span. Israeli law does not at present provide guidelines as to the proper principles or purposes of sentencing, but leaves this to the discretion of the individual judges. A judicial commission has recently recommended the establishment of base-sentences by legislation, and the Israeli Ministry of Justice is in the process of developing a concrete legislative proposal. The base-sentence for a particular offence is defined as the sentence a judge ought to impose in the absence of mitigating or aggravating factors.

The aim of Judge's Apprentice is to provide support for the sentencing of offenders found guilty of two types of crimes: rape or robbery. The system is based on the following model: (1) Domain experts are used to discover relevant factors. This corresponds to the æshopping listÆ of the previous section. (2) The existence of base sentences is assumed. (3) Setting out from the base sentence, the system weighs aggravating (which increase the sentence) and mitigating (which decrease the sentence) factors. For example, if the victim of a rape case is young, then this is an aggravating factor whilst the fact that the perpetrator is young is a mitigating factor. Similarly, if a criminal performs an armed robbery, then the fact that she was armed is an aggravating factor, whilst lack of prior criminal convictions is a mitigating factor.

From our study of property division in Australian Family Law and sentencing in Israeli Criminal Law we note that both domains involve numerical decision making, but whereas the latter has norms the former has very few norms, but it does have well determined attributes or factors.

2.3 Identifying different forms of discretionary domains

Following on from the research of [16] on open textured and bounded legal predicates, we have noticed that there are three

[2] Judges using the asset by asset approach in devising an equitable property order on divorce litigants carve up the property asset by asset whereas those using the global approach lump all assets together and award a percentage of the total to both parties.

distinct ways in which discretion can be exercised in legal domains. We have defined these concepts as, narrow discretion, bounded discretion and unfettered discretion (see also [10] and [18]).

- **Narrow discretionary domains** have well enunciated norms which have been formulated from legislation, cases and/or legal opinions. Judges have the ability to deviate from these norms (but not by significant amounts). According to our sentencing model the domain in Israel is narrow discretionary because of the assumption of the existence of base sentences. Israeli judges use their discretion to deviate from these base sentences û given circumstances such as violence, damage to the victim or the record of the offender. Property division following divorce, as exercised in many countries (such as the United States of America and Israel, but not Australia) is narrow discretionary. Judges commence with a 50/50 split to each party and then make deviations because of certain factors such as the needs of children and the health of each party.

  Such domains are bounded and primarily well definedû in that we have a specific list of factors which lead to deviation from the norm. We can use case based reasoning on landmark cases to model such domains.

- **Bounded discretionary domains** have no norms. Legislation or cases specify what factors judges must take into account. However judges are not told what weight to attribute to the factors. An example of such a domain is the percentage split distribution of common pool property in Australian Family Law. The factors to be taken into account are specified in Section 75(4) of the Act. However, nowhere are judges instructed on the significance of each of these factors

  The predicates in bounded discretionary domains are open textured but bounded. We can hence model such domains through the use of Knowledge Discovery from Databases techniques.

- **Unfettered discretionary domains** have no norms and judges are not even told what factors must be taken into account in reaching a decision. Such domains are both open textured and unbounded and we do not believe it is wise to model them.

  An example of unfettered discretion is the determination of the custody of children in Australian Family Law. According to the Family Law Act (1975) the only factor to be taken into account is *the paramount interests of the child*. Following considerable litigation and uncertainty the Australian Federal Parliament made minimal attempts to define what are the paramount interests of a child. They did this by identifying in the legislation factors such as education, health, a child's relationship with both parents, and the need to keep siblings together. But there is no clear list of factors. Indeed it is much easier to describe what is not in a child's best interests (for example sexual abuse, violence, emotional abuse) than what is in a child's best interests.

  We should stress that it is not the case that all countries give Family Court Judges unfettered discretion in deciding the custody of children. In some countries religious education is given prominence, in others it is assumed that mothers are best able to care for young children and daughters.

Recent decisions regarding the distribution of property in Australian Family Law, are fortunately uniform. For a number of years after the inception of the Family Law Act(1975), Family Court Judges were not required to explain their decisions in great detail. In *Bonnici vs Bonnici*[3] (1991) the Full Court of the Family Court of Australia emphasised that trial judges must provide comprehensive reasons for judgements. As a consequence, each case now decided includes detailed reasons for the property decision reached. Thus we have a uniform case base (or training set) and can use knowledge discovery techniques.

3 SplitUp and Judge's Apprentice: two examples of numerical discretionary legal knowledge based systems

SplitUp is a hybrid rule-based/neural network system which provides advice upon the distribution of property following divorce in Australia. The task of determining what property a Family Court judge may distribute was determined to be ruleûbased and implemented using directed graphs. A hierarchy of ninety four factors relevant for a percentage split prediction were identified with the help of domain experts. The way some factors combine was later learnt by machine learning algorithms known as neural networks. The way other factors combined was modelled with rules that derived from expert heuristics.

Deciding the associated weight for each factor is a very important problem. Whist the SplitUp project has focussed on using neural networks for this task [21] reports on the use of rule induction systems whilst [15] investigate the use of linear regression.

Neural networks provide no explanation for their answers. The SplitUp system utilises a knowledge representation that is based on the argumentation theory of Toulmin [17]. Toulmin argument structures provide a representation which allows for a separation of inferencing from explanation. A claim can be inferred from data values using a neural network or a rule set or, conceivably any other inferencing method. An explanation is

---

3 15 Fam LR 138 (1991)

generated by reproducing the data, warrant or backing and is performed after, and independently from the claim inference.

Judge's Apprentice (as indeed does any case based reasoning system) looks for similarities between a new case and old cases stored in the case base. The most relevant precedents are retrieved through the use of a similarity measure. A major focus of Judge's Apprentice is to compute new sentences. Judge's Apprentice does this by (i) finding most relevant case (ii) using the base sentence and the sentence in the retrieved to evaluate a sentence for the new case.

HYPO [2] and CATO [1] provide a different way of representing and harnessing the interaction among factors in a case, than does the Judge's Apprentice. HYPO and CATO also apply to a new case the interactions among factors in old cases, but they use a symbolic, argumentation-oriented approach.

4 Computation of a Sentence by the Judge's Apprentice

4.1 Basic Assumptions

We assume that our domain knowledge is defined as follows:

(1) Each type of crime has a base-sentence B. The base-sentence is defined as the sentence a judge would give, if there were *no* mitigating or aggravating circumstances at all.

(2) Legal knowledge about sentencing is expressed in a discrimination tree (as in the book of Kolodner [5]) where each node defines a mitigating or an aggravating factor. Most of these factors are general and common to all crimes, some relate to specific crimes only.

(3) Each node has an associated weight W. W is defined as the percentage to be added to (or subtracted from) the base-sentence if the particular factor represented by the node was the *only* mitigating or aggravating factor.

(4) Our information about a new case consists of the set of relevant factors from the discrimination tree with their associated weights Wi. These factors are the *retrieval indexes* of the case.

(5) We assume the existence of a uniform case-base: Each case having been selected as having a representative sentence (presumably by some judicial commission).

While it appears that the experts agree about what the factors of the discrimination tree should be [11], they do not agree about the values of the weights and the base-sentences. This is one of the reasons for sentencing disparity. So far we have built only a prototype, and have not worried about this issue.

We have already noted that the Ministry of Justice in Israel is preparing a legislative proposal defining base-sentences for all crimes in the criminal law. Weights could also be defined by legislation or by a judicial commission, but we are aware of the essentially political problems that would arise. These problems will of course also be present in the selection of the landmark cases.

It should be stressed that the above assumptions and the computational model itself as it is described below, in no way imply that judges actually pass sentence according to that model. For judicial sentencing models see Lovegrove [7] and von Hirsch and Ashworth [18]. The computer program based on our model should simply be considered as one of several possible implementations of the case-based paradigm. This implementation model is straightforward and intuitively easy to understand.

4.2 Approach using only the New Case

A primitive approach to sentence computation would simply be to add the weights of the relevant factors to the base-sentence:

$$\text{Sentence}_{\text{New case}} = \text{Base-sentence} * [\, 1 + \Sigma\, W_i\,(\text{New case}) / 100\,]$$

Obviously this approach is wrong: Sentences cannot be computed in a simple additive manner. Human discretion means that the judge considers some of the relevant factors together, and decides on some non-explicit and undefinable interaction of their weights.

4.3 Approach using Retrieved Case

Assume the system has retrieved the nearest case (in some similarity sense) to the new case from the case-base. One can now argue that the interaction among factors in the old case should be applied in the new case, even though not all the factors of the two cases are identical or even near-neighbours in the discrimination tree. For the old case is retrieved precisely because it is the nearest of all cases in the case-base to the new case. We therefore adjust the sentence of the retrieved case by a coefficient of proportionality, and obtain:

$$\text{Sentence}_{\text{New case}} = \text{Sentence}_{\text{Retrieved case}} * (\Sigma\, W_i\,(\text{New case}) / \Sigma\, W_j\,(\text{Retrieved case}))$$

4.4 Approach using Retrieved Case and Base-Sentence

Over time sentences will change, as both base-sentences and the relative weights of the factors change. Following detailed discussions with judges and other domain experts, it is our impression that the relative weights of the factors change far more slowly than the base-sentence. We therefore believe that the derivation of a sentence should depend on the base-sentence in an explicit manner. Furthermore, the base-sentence is common to both the new and the retrieved case, and the adjustment coefficient should therefore not change that term, as happens in the formula of the preceding section.

For any case let us define æparticularÆ weights V_i for all factors relevant to the case in the following manner: These particular weights are themselves unknown and differ from case to case, but their *combination* result in the actual sentence for that particular case:

$$\text{Sentence} = \text{Base-sentence} * [\ 1 + \Sigma\ V_i / 100\ ]$$

Given a new case and a retrieved similar case we know the sum of the particular weights for the retrieved case. Our idea is to use the same combination for the new case, with an adjustment of proportionality, for the sum only. We thus obtain:

$$\text{Sentence}_{\text{New case}} = \text{Base-sentence} * [\ 1 + \Sigma\ V_i(\text{Retrieved case}) * \Sigma\ W_i(\text{New case}) / \Sigma\ W_i(\text{Retrieved case})\ ]$$

We have tried several other approaches . Examples include dividing the factors of the new case into three sets: (1) Factors identical to factors in the retrieved case, (2) factors similar to factors in the retrieved case and (3) factors not identical and not similar to any factor in the retrieved case. It appears that the results are quite insensitive to the particular form of the formula, so we decided on the above one, which is simple and theoretically intuitive.

It is interesting to note that it will be possible to use a retrieved case, even if its sentence is not up-to-date. What is important is the way its factors combine.

4.5 A Sample Case using the JudgeÆs Apprentice

Consider for example the following case taken from the Tel Aviv District Court: A man of 28 is found guilty of raping a woman of 24. Whilst they previously had consensual sex for some time (four years), the woman eventually broke off the relationship and refused to have sex with her ex-partner. The man admitted committing the rape and had no prior criminal record. The woman understood why the man had committed the rape, but she did not forgive him. The probation officer informed the judge that a prison sentence would psychologically harm the offender.

The closest retrieved case concerned a man of 25 and woman of 23 who previously had a consensual sexual relationship for a short period of time (3 months). When the woman ended the relationship the man forced sex upon her. The woman forgave the man for the rape and indeed the civilian police encouraged the couple to marry. The man had a prior criminal record, but the offences were minor. The man had an honourable service record.

The system indicates (as a default) the seven factors with the heights weights when comparing the new case (NC) and the retrieved case (RC). Any number smaller than seven would be too small for comparison. Furthermore the number seven is accepted as the number of items that the average person is able to remember without apparent effort Miller [8]. If desired. the user may retrieve a comparison of the full set of factors relevant to the two cases.

[12] have discussed feature selection in the SplitUp system. They applied feature selection techniques using genetic search to the data used to determine percentage split in the Split Up system . Genetic algorithms were used to determine which attributes are essential to model when distributing marital property. Their research shows a more accurate prediction can be made when using 16 of the 94 variables. An interpretation of this result is that the other 78 attributes are rarely used by Family Court judges when distributing property.

For the above case, the seven most important comparisons between NC and RC were:

<u>Main Identical Details</u>

Detail

1. Couple had prior consensual sexual relations

2. Victim was young

3. Offender confessed

4. Offender was young

Main Similar Details

NC Detail	RC Detail	Source of Detail	Comparison
1. Offender had no prior criminal record	Offender had minor criminal record	Prior convictions	More lenient
2. Woman understood man's actions but did not forgive him	Woman understood man's actions and did forgive him	Victim statements	Less lenient
3. Offender knew victim for 5 years	Offender knew victim for 3 months	Prior relationship	Less lenient

Main Details in NC but not in RC

NC Detail	Comparison
1. Probation officer claimed prison sentence would psychologically harm offender	More Lenient

Main Details in RC but not in NC

RC Detail	Comparison
None	

The sentence in RC was 24 months. For NC, Judge's Apprentice took into account the above mentioned aggravating and mitigating factors and recommended a decreased sentence by 58%. This resulted in a recommended sentence of 10 months. It claimed a confidence level of 92%. The District Court actually gave the offender a sentence of 8 months. NC went on appeal to the Supreme Court, where the offender received a conditional (suspended) sentence for unrelated reasons.

5. Evaluating SplitUp and the Judge's Apprentice

An explanation of a suggested sentence range in the Judge's Apprentice is exactly the answer: namely the base sentence (B), closest case (RC), the differences between the new case (NC) and RC and reasons for different sentences in RC and NC.

Both Judge's Apprentice and SplitUp provide decision support for arriving at numerical answers in discretionary legal domains. SplitUp uses commonplace rather than landmark cases. So there is no explicit determination of closest cases. Admittedly, when learning weights on which to make a decision from a Neural Network, the closer an old case (one in the training set) is to the New Case the more influence the Old Case will have in determining a decision for the new case. Thus whether using Neural Networks or Case Based Reasoning, a measure of closeness is importance. For Case Based Reasoning this measure is explicit, for Neural Networks it is implicit.

Proper testing and validation of any system is important for determining the accuracy, completeness and performance of a system [9]. Judge's Apprentice was successfully trialed before four senior Israeli legal experts who found the system both intelligent and interesting. Three of the experts felt the system could prove invaluable to judges. The system was tested on a non-typical database comprising fifty-four precedents. A test was carried out for each precedent comparing the sentence computed by the system and the sentence that was actually passed. The test was carried out according to the "leave one out" method, in which each of the cases was tested as if it was a new case with each of the other cases as precedents for it. A sentence for NC (constructed by the system in accordance with the retrieved precedent) was compared to the actual sentence decided by the judge. Success was defined if the predicted sentence was within 33 1/3 % of the actual. Given that the database was small, excellent results could not be expected. Where the precedents retrieved were of high similarity, the success rate was 76%. Where the cases were of medium similarity, the success rate was only 47%. This indicates that for useful legal discretionary support systems to be constructed, sufficiently large case bases must be developed. The SplitUp project has also noticed this fact.

Whereas Judge's Apprentice has only been tested by judges and academics, SplitUp has been widely tested by various categories of users. SplitUp [13] has been evaluated in five distinct ways. 1. Domain expert assessment of both the content and structure of the SplitUp knowledge base and the problem solving strategy employed in SplitUp; 2. Comparison of predictions made by SplitUp with those made by eight lawyers on the facts from the same three cases; 3. The use of SplitUp on a new trial case recently concluded in the Family Court of Australia. 4. Feedback from users in four different categories using SplitUp predictions and explanations û by seven lawyers, four registrars of the Family Court of Australia, three judges of the Family Court of Australia and five others. Preliminary results show that the system is particularly valuable for mediators and users.

In Judge's Apprentice consideration of retrieved cases next in order of closeness may lead to contradictory advice, even if the case-base is uniform. This is no problem. Legal experts are accustomed to reason with cases yielding conflicting conclusions. The judge has to decide which advice to follow (if at all). The SplitUp project developed measures of error (for example a 5% difference either way was considered insignificant) and needed to deal with contradictory cases û otherwise the KDD techniques would not learn any results. The decision by the SplitUp project [14] was to exclude all contradictory cases. Alternatives which were considered (but not adopted) were a) only use those cases which formed a majority opinion; b) use all cases.

6. Current research on discretionary legal support systems

Both SplitUp and Judge's Apprentice support: (i) enhanced consistency in decision-making; (ii) community understanding of the domain; and (iii) advanced dispute resolution.

Alternative dispute resolution û In the SplitUp system we have used knowledge discovery techniques to determine how Australian Family Court judges use their discretion to distribute marital property. While Split_Up can be used to determine ones BATNA (Best Alternative to a Negotiated Agreement) for a negotiation, it does not model the negotiation process itself.

The SplitUp system informs both litigants what they would be expected to be awarded by a court if their relative claims were accepted. It thus advises them upon their BATNA. The system also gives the litigants relevant advice as to what would happen if some, or all of their claims were rejected. They are able to have dialogues with the SplitUp system about hypothetical situations which would support their negotiation. Both litigants then have clear ideas about the strengths and weakness of their claims.

Judge's Apprentice can be used to inform defendants to a criminal case as to the likely penalty they will receive if convicted of the crime, and what reduction in sentence will ensue if they plead guilty. The process of plea bargaining is illegal in some countries (such as Australia and the United Kingdom). But where legal, the process can free up courts, saving the community money and leading to shorter waiting periods before trials are heard.

Developing Judge's Apprentice into a real world system — The software, both engine and user interface, are acceptable to users of the legal profession. The only prerequisite is a uniform case-base acceptable to the judicial authorities. Owing to the public criticism of the present sentencing disparity, it may be politically feasible to establish a judicial commission, empowered to select cases for this case-base.

Commercialising the Split Up system — The SplitUp research has shown it is feasible to build legal decision support systems that learn from commonplace cases. However the current version of SplitUp is a research prototype. To build a decision support system that is widely used by legal practitioners the project must: a) develop a much better user interface; b) ensure the system is more robust; c) add hundreds more recent commonplace cases to the system; d) perform detailed evaluation experimentation. Recently, the Australian Research Council, through its SPIRT (Strategic Partnership in Research and Training) grants has given the SplitUp project a large three year grant. Together with Victorian Legal Aid and Phillips and Wilkins Solicitors the project will receive $500,000. to conduct research and eventually build a commercial system using fuzzy logic. At the conclusion of this project, when the case base is increased ten-fold, evaluation is rigorously undertaken and alternative KDD techniques explored, we expect to have a robust commercial system.

7. Conclusion

Except in taxation law and social security law, few legal knowledge based systems have been built that offer numerical advice. In our attempt to build intelligent legal knowledge systems which provide numerical advice in discretionary legal domains we have developed the notion of three forms of discretion: narrow discretion, bounded discretion and unfettered discretion. Through the use of examples we illustrated that the first form of discretion can be modelled using case-based reasoning, the second form can be modelled using rule based reasoning whilst the third form should not be modelled.

The sentencing of Israeli criminals is a narrow discretionary domain. Judge's Apprentice is a case-based reasoner which recommends ranges of sentences for convicted rapists and robbers. Property division in Australian Family Law is a bounded discretionary domain. SplitUp uses Knowledge Discovery from Databases to learn what percentage of marital property the husband will receive.

We conducted a detailed examination of Judge's Apprentice and contrasted it with the SplitUp system with regard to reasoning, explanation, evaluation and dealing with contradictory cases. Systems such as Judge's Apprentice and SplitUp are leading to enhanced public confidence in judicial discretion making as well as offering support for dispute resolution.

8 References

[1] Aleven, V. and Ashley, K.D. 1995. Doing things with factors. *Proceedings of Fifth International Conference on Artificial Intelligence and Law*, ACM:185-194.

[2] Ashley, K. D. 1991. *Modelling legal argument: Reasoning with cases and hypotheticals*. Cambridge: MIT Press.

[3] Hart, H.L.A. 1961. *The Concept of Law*. Oxford: Clarendon Press.

[4] Hassett P. 1993. Can Expert System Technology Contribute to Improved Bail Conditions. *Int. J. of Law and Information Technology* 1: 144-188.

[5] Kolodner J. 1993. *Case-Based Reasoning*. San Mateo, CA: Morgan Kaufmann Publishers.

[6] Kovacs, D. 1992. *Family Property Proceedings in Australia*. Butterworths, Sydney.

[7] Lovegrove A. 1997. *The Framework of Judicial Sentencing*. Cambridge, UK: Cambridge University Press.

[8] Miller G. A. 1956. The Magical Number Seven, Plus Minus Two: Some Limits on our Capacity of Information. *Psychological Science*, 63:81-97.

[9] O'Leary, T. J., Goul, M., Moffitt, E. and Radwan, A. E. 1990. Validating Expert Systems, *IEEE Expert* 5:51-58.

[10] Pattenden R. 1982. *The Judge, Discretion, and the Criminal Trial*. Oxford, UK: Clarendon Press.

[11] Schild U. J. 1998. Decisions Support for Criminal Sentencing. *Artificial Intelligence and Law*. 6(4): 151-202.

[12] Skabar, A., Stranieri, A. and Zeleznikow, J. 1997. Using argumentation for the decomposition and classification of tasks for hybrid system development. in Kasabov, N., Kozma, R., Ko, K., O'Shea, R., Coghill, G., and Gedeon, T. (eds) *Progress in Connectionist Based Information Systems. Proceedings of the 1997 International Conference on Neural Information Processing and Intelligent Information Systems*. Springer-Verlag. Singapore: 814-818.

[13] Stranieri, A. 1998. *Automating legal reasoning in discretionary domains* (PHD Thesis, La Trobe University, Melbourne, Australia)

[14] Stranieri, A. and Zeleznikow, J. 1998. A re-examination of the concepts of open texture and stare decisis for data mining in discretionary domains. *Proceedings of Eleventh International Conference on Legal Knowledge Baed Systems,*, Koniklijke Vermand, Gronigen, Netherlands: 101-111.

[15] Stranieri, A. and Zeleznikow, J. 1999. Evaluating legal expert systems. To appear in *Proceedings of Sevenfh International Conference on Artificial Intelligence and Law*, ACM.

[16] Stranieri, A., Zeleznikow, J., Gawler, M. and Lewis, B. 1999. A hybridûneural approach to the automation of legal reasoning in the discretionary domain of family law in Australia. To appear in *Artificial Intelligence and Law*
.

[17] Toulmin, S. 1958. *The uses of argument*. Cambridge: Cambridge University Press.

[18] Von Hirsch A. and Ashworth A. (ed.) 1998. *Principled Sentencing* (2nd edn). Oxford, UK: Hart Publishing.

[19] Waterman, D.A. and Peterson, M. 1980. Rule-based models of legal expertise. In the Proceedings of the *First National Conference on Artificial Intelligence*, 272-275, Stanford University: AAAI.

[20] Yearwood, J. 1997. Case-based retrieval of Refugee Review Tribunal Text Cases. *Proceedings of Tenth International Conference on Legal Knowledge Baed Systems,*, Koniklijke Vermand, Gronigen, Netherlands: 67-83.

[21] Zeleznikow, J., Stranieri, A. and Gawler, M. 1996. Split-Up : A legal expert system which determines property division upon divorce. *Artificial Intelligence and Law* 3:267-275.