AI and Law:
How to Get There from Here

L. Thorne McCarty
Computer Science Department
and
Faculty of Law
Rutgers University
New Brunswick, NJ 08903, USA

March 1990

Abstract

This paper offers a survey of the current state of AI and Law, and makes recommendations for future research. Two main areas of investigation are discussed: the practical work on intelligent legal information systems, and the theoretical work on computational models of legal reasoning. In both areas, the knowledge representation problem is identified as the most important issue facing the field.

To appear in: Ratio Juris

This paper was originally presented at the International Conference on “Expert Systems in Law”
Bologna, Italy
May 3-5, 1989

Copyright © 1990 L. Thorne McCarty
1 Introduction

The very first international conference on the applications of artificial intelligence to law was held in Swansea, Wales, from September 17-27, 1979 [43]. This was approximately ten years ago, and so it is fitting now to take stock of the field and outline some possible directions for future research.

Indeed, 1989 has seen a veritable celebration of the tenth anniversary of the Swansea meeting, with so many conferences on the subject scheduled during the course of the year. In addition to the conference on “Expert Systems in Law” here in Bologna, in May, there is the “Second International Conference on Artificial Intelligence and Law” in Vancouver, in June, and the “Third International Conference on Logica, Informatica, Diritto” in Florence, in November. Note that the original Swansea meeting spanned eleven days(!), but the Bologna meeting lasts only three days. And yet a cursory glance at the program from Bologna shows that there are four times as many participants and three times as many papers presented in Bologna as there were in Swansea. Presumably, we are much more efficient in communicating our ideas now than we were then.

In this paper, I will argue that we have not advanced very far at all in these past ten years, but we have learned a lot about where to direct our efforts in the future. These are very subjective judgments; others will surely disagree with both the diagnosis and the cure. I will primarily describe my own work, and my own research programme, and I realize that this perspective may be biased. But a touch of dogmatism can be useful in provoking a debate that might ultimately lead to a better understanding of the issues. We have seen this already in the thoughtful essay by Trevor Bench-Capon on
“Deep Models, Normative Reasoning and Legal Expert Systems” [4], which is discussed in Section 2 below. Perhaps the present paper will also contribute to this dialogue.

The field of AI and Law has always had two distinct motivations: practical and theoretical. On the practical side, we are trying to build “intelligent legal information systems” [28, 33], systems that can assist both lawyers and nonlawyers in their interactions with both legal and nonlegal rules. On the theoretical side, we are trying to gain a better understanding of the process of legal reasoning and legal argumentation, using computational models and techniques. I have always maintained that these two objectives should be carefully distinguished from one another, since the criteria for success are so different in each case. Accordingly, I will treat them separately in the present paper. Section 2 discusses intelligent legal information systems, and Section 3 discusses computational models of legal reasoning. Section 4 contains a few brief concluding remarks.

2 Intelligent Legal Information Systems

Today, there are many “rule-based expert systems” in law, each developed at least to the point of a research prototype. Recent examples include the work of Sergot, et al. [52]; Biagioli, Mariani, and Tiscornia [6]; Greenleaf, Mowbray and Tyree [21]; Smith and Deedman [56]; Oskamp, et al. [44]; Sherman [53]; and Weiner [62]. Although the technologies vary somewhat, these systems all have the following characteristics: They do backward-chaining inference from a specified goal; they ask questions to elicit information from the user; and they produce a suggested answer along with a trace of the supporting legal rules. The work of Richard Susskind is particularly instructive in this regard. Susskind has argued that a restricted form of backward-chaining propositional inference is sufficient for many practical legal tasks, citing a consensus among diverse jurisprudential theorists [58], and he has backed up this claim in his work with Philip Capper on an expert system for the English law of latent damages [59]. It is interesting to note that the Susskind/Capper system, although extremely sophisticated in its legal analysis, uses a very simple (almost trivial) expert system shell that is available commercially for less than $1000.
It thus appears that we can build practical expert systems in law without any further work on artificial intelligence! But let us examine what these existing systems cannot do. I will use the classification suggested in my earlier papers [28, 33]:

(1) Legal analysis systems. To build a legal analysis system with current expert systems technology, it is necessary to formulate a very narrow goal: “Is X a British citizen?” “Is the cause of action for Y barred at time T?” Without such a goal, it is impossible to collect information from the user. Even with such a goal, the questions posed by the system are often tedious, and the problem gets worse as we try to accommodate a broader or more abstract range of legal issues. Instead, we would like to allow a user who already knows something about his domain to tell his own story, using a rudimentary natural language interface, perhaps, and then have the system perform a partial analysis before it queries for further information. In the CAI literature, this is referred to as a “mixed initiative” dialogue [10].

(2) Legal Planning Systems. To date, there have been no true legal planning systems. Those systems that have provided advice on tax planning, such as TaxAdvisor by Michaelsen [41] and EPS by Schlobohm and Waterman [51], have simply reduced the planning problem to a diagnosis problem, thereby forcing it into the mold of an expert system shell. But there is additional work on planning and design in AI [11, 60] which matches the process of legal planning and design quite closely. In this paradigm, we first develop a language in which we can describe the desired artifact (e.g., an estate plan) and state the constraints on an acceptable solution (e.g., minimal tax consequences). We must then solve a massive search problem, but there are various techniques (hierarchical decomposition, incremental refinement, etc.) that can make the search manageable in particular situations. Applied to the law, these techniques would give us a true legal planning system.

(3) Conceptual Legal Retrieval Systems. This is an area pioneered by Carole Hafner [22, 23]. The basic idea is to build up a data base containing the essential information about a collection of cases: the facts, the applicable legal rules, the alternative analyses, etc. Retrieval is then guided by a search for matching facts, or matching rules, or matching patterns of analysis. For an interesting proposal to organize this search around an explicit theory of legal argument, see [12]. One of the major problems in this area, long recognized, is the difficulty of constructing a data base of realistic size and
scope. Jon Bing has recently proposed a compromise solution, in which the existing full-text key-word retrieval systems (*LEXIS*, *WESTLAW*, etc.) are retained, but an expert system is interposed between the user and the query [7]. Bing calls this a *norm-based thesaurus*, since the expert knowledge in the system would be embodied in a representation of the statutory norms.

Conventional rule-based expert systems may play some role in each of these applications, but it should be clear that more sophisticated techniques are needed, in general. Moreover, our ultimate goal should be a hybrid system that combines all three of the components listed above, and perhaps more. The legal analysis and planning systems would be more useful if they could provide direct access to the cases; the conceptual retrieval system would be more powerful if it could use the same rules as the analysis and planning systems. I have argued in my earlier papers [28, 33] that the key to all of this is the construction of a “deep conceptual model” of the legal domain.

Although some commentators have expressed puzzlement about the meaning of the term “deep conceptual model” [58, pages 149-155], the basic idea is easy to state. There are many common sense categories underlying the representation of a legal problem domain: space, time, mass, action, permission, obligation, causation, purpose, intention, knowledge, belief, and so on. The idea is to select a small set of these common sense categories, the ones that are most appropriate for a particular legal application, and then develop a *knowledge representation language* that faithfully mirrors the structure of the selected categories. The language should be formal: it should have a compositional syntax, a precise semantics and a well-defined inference mechanism. The semantic interpretation of the common sense categories should be intuitively correct, that is, it should generate exactly those entailments that ordinary people (and ordinary lawyers!) generate in similar situations. The inference mechanism for the language should be complete and sound, in principle, but, in practice, completeness and soundness would often be sacrificed for computational tractability, just as they are in ordinary human (and ordinary legal!) reasoning.

An example of such a knowledge representation language is my own *Language for Legal Discourse*, or *LLD*, which is described in [34]. *LLD* has facilities for the representation of states, events, actions, and various modalities over actions such as permission and obligation. There are similar facilities in the *Event Calculus* of Kowalski and Sergot [25], although Kowalski seems
to have taken a principled stance against the explicit representation of the deontic modalities [24]. LLD also provides a systematic treatment of sorts and subsorts (e.g., an ‘Actor’ can be a ‘Person’ or a ‘Corporation’), and it includes both count terms and mass terms (e.g., ‘Person’ is a count term and ‘Stock’ is a mass term). For both technical and philosophical reasons, the language is based on intuitionistic logic rather than classical logic. I have argued elsewhere that an intuitionistic semantics offers distinct advantages for a logic programming language [31, 32], and these advantages are inherited by the action language and the deontic language in LLD [29, 30]. In addition, intuitionistic logic provides a framework for the representation of default rules [36], and it is essential for the theory of prototypes discussed in Section 3 below. Tom Gordon has also stressed the importance of nonmonotonic reasoning in the law [19], and his proposals for knowledge representation languages [18, 20] have some similarities to my own.

The design of LLD was originally motivated by my work with Sridharan on the TAXMAN II project [38, 39], and it was subsequently influenced by the work of two of my students on the Uniform Commercial Code [8, 40]. More recently, LLD has played a key role in my work with Dean Schlobohm on estate planning [50]. Our proposed system, EPS II, is not a diagnosis system like EPS [51], but a true legal planning system which attempts to construct an estate plan by reference to a “deep conceptual model” of the legal domain. For example, we formulate the estate planning problem for inter vivos gifts as follows. A client usually has three kinds of goals: (i) providing for beneficiaries, while (ii) retaining control over assets and (iii) avoiding taxes as much as possible. But these goals are mutually incompatible, and it is the job of an estate planner to construct a trust that strikes an appropriate compromise between (i), (ii) and (iii). What is a trust? It is a legal instrument which places the trustor’s assets under the control of a trustee, and stipulates that the trustee is permitted, or forbidden, or obligated to take certain actions with respect to these assets, under certain specified conditions. The Internal Revenue Code then characterizes the trust assets as either included or not included in the client’s gross estate, depending on the powers granted, respectively, to the trustor and the trustee, and this determines the tax consequences of the plan. For our purposes, the important point here is that the mere statement of the estate planning problem and the description of the space of possible solutions requires a language like LLD,
with facilities for count terms, mass terms, states, events, actions, and all of the deontic modalities. There seems to be no way to build a true legal planning system in this domain without such a language.

Trevor Bench-Capon has argued in [4] that intelligent legal information systems for most practical applications can be built without deep conceptual models and without the use of deontic logic. Part of his argument has already been considered here: It is true that there are many applications for systems like Susskind and Capper’s Latent Damage Analyst [59], and the remaining question is simply how much additional benefit might be realized from the more sophisticated analysis/planning/retrieval systems advocated above. However, Bench-Capon also criticizes the attention given by some researchers in AI and Law to normative reasoning, and here he misinterprets the role of deontic logic in my own work. Deontic logic is needed, not because laws state general norms, but because the deontic modalities appear prominently among the common sense categories that constitute the lawyer’s world. In our estate planning system, for example, we do not explicitly represent the obligation of a citizen to pay taxes, since this is taken for granted when we are trying to develop an estate plan, but we do explicitly represent the characteristics of a trust, and this requires us to represent the obligations of the trustee. The deontic modalities have no special status here. They are simply one of the many common sense categories that we need to incorporate into our knowledge representation language in order to build a practical system.

Bench-Capon also suggests that those of us who are interested in “deep conceptual models” are really interested in legal theory, not just legal practice. This is true, of course, and it is the subject of the following section.

3 Computational Models of Legal Reasoning

Although most of the work on AI and Law today is oriented towards the development of practical systems, there is a small group of researchers who are primarily interested in theoretical questions: How much of legal reasoning can be reduced to reasoning with rules? Is this rule-based component significant, or trivial? How is it possible to reason with cases at all? Are
legal concepts just like ordinary common sense concepts, or do they have special characteristics? Is it possible to develop a computational theory of legal argument? The researchers who have investigated these questions include: Anne Gardner [16]; Edwina Rissland and her students, Kevin Ashley [45, 1] and David Skalak [46, 54]; Karl Branting [9]; and Keith Bellairs [3]. In addition, researchers such as Richard Susskind [58] and J.C. Smith [56], who have primarily built practical systems, have also been deeply concerned with the jurisprudential foundations of the field.

I will not attempt to summarize this work in the present paper. Instead, I will summarize and criticize my own prior work on the **TAXMAN II** project, and suggest that some of these criticisms apply as well to more recent theoretical research on AI and Law.

Almost ten years ago, Sridharan and I proposed a theory of legal reasoning in hard cases [27, 38, 39]. We began by emphasizing the following three points, which should be familiar to most lawyers:

1. Legal concepts cannot be adequately represented by definitions that state necessary and sufficient conditions. Instead, legal concepts are incurably “open-textured”.

2. Legal rules are not static, but dynamic. As they are applied to new situations, they are constantly modified to “fit” the new “facts”. Thus the important process in legal reasoning is not theory application, but theory construction.

3. In this process of theory construction, there is no single “right answer”. However, there are plausible arguments, of varying degrees of persuasiveness, for each alternative version of the rule in each new factual situation.

The first of these points has been thoroughly discussed by Anne Gardner [16], and seems to be generally accepted by researchers in AI and Law. The second point is less common, but it is related to the constructive approach to legal decisions proposed by Herbert Fiedler [15] and to the rule-based representation of open-texture in law proposed by Trevor Bench-Capon and Marek Sergot [5]. The third point, of course, has been thoroughly debated by legal philosophers for many years as part of the response to Ronald Dworkin’s
thesis [14]. Sridharan and I adopted this third point primarily as a methodological guideline: Since lawyers are more likely to agree on what counts as a plausible argument in a case than to agree on the appropriate outcome, we decided that it would be more fruitful to develop a theory of legal argument than to develop a theory of correct legal decisions.

This was the framework in which we worked. The specific theory we proposed was based on a representation of legal concepts by means of prototypes and deformations. Legal concepts have three components, we suggested: (1) an (optional) invariant component providing necessary conditions; (2) a set of exemplars providing sufficient conditions; and (3) a set of transformations that express various relationships among the exemplars. These three components are then refined further, for most concepts, so that one or more of the exemplars is designated as a prototype and the remaining exemplars are represented by a set of transformations, or deformations, of the prototypes. In this model, the transformations induce a partial order on the set of exemplars corresponding to the typicality gradient observed by psychologists in the study of human categorization [47, 55], and the application of a concept to a new factual situation automatically modifies the definition of the concept itself, as required by Levi’s classical account of legal reasoning [26]. This was our response to the first two points noted above. In addition, in response to the third point, we were able to show that the arguments of lawyers and judges in a series of early corporate tax cases could be explained very well by the theory of prototypes and deformations. Our principal example was Eisner v. Macomber, 252 U.S. 189 (1920), an early stock dividend case, in which the arguments of Justice Pitney and Justice Brandeis took the form of a sequence of transformations from precedent cases through hypothetical cases to the factual situation of Macomber. It is important to note that these “explanations” were hand simulations. The TAXMAN II theory was partially implemented by Donna Nagel in her thesis [42], but a full implementation was never attempted.

I still believe that the TAXMAN II theory is qualitatively correct. But there were two major problems with our earlier work. First, the theory makes enormous demands on our knowledge representation language. To see this, it is sufficient to note that a transformation is a syntactic operation, and for such an operation to be meaningful it must correspond to the significant semantic relationships in the legal domain. However, the frame-based language
in which we implemented the *TAXMAN I* system [37, 57] did not have an adequate semantic foundation, and this meant that a full *TAXMAN II* implementation would have been entirely *ad hoc*. The second problem involves the theory of prototypes and deformations itself. What determines the choice of a prototype? What are the criteria for constructing transformations? It was clear that the set of transformations had to be tightly constrained, or else anything could be “transformed” into anything. But what was the source of these constraints? Much of my work since 1982 has been devoted to finding solutions to these two problems.

I believe that the first of these problems has now essentially been solved by my *Language for Legal Discourse* [34], which was discussed in Section 2. It is no accident that the common sense categories embodied in the current version of *LLD* are just those categories that we need for an initial representation of corporate tax law: count terms, mass terms, states, events, actions, permissions, obligations. This was one of the reasons that *LLD* was invented. Other categories will surely be needed later, if we wish to develop a more sophisticated analysis of the tax code: purpose, intention, knowledge, belief are prime examples. Why do I insist that *LLD* is a solution? One of the main features of the language is the close correspondence between its surface syntax and its deep semantics, and this corrects the first deficiency in our earlier work. Syntactic transformations now map directly onto significant semantic relationships.

The second problem is not completely solved by *LLD*, but the necessary tools are now available. I remarked above that the theory of prototypes and deformations requires a set of tight constraints on transformations. In our earlier papers [38, 39], Sridharan and I noted that these constraints seem to be related to a sense of “conceptual coherence”. But what does that mean? My conjecture now is that conceptual coherence can be explained, at least partially, by an analysis of the computational tractability of the inferences that we normally want to make in a language with the features of *LLD*. Since *LLD* is so rich and expressive, the proof procedures for the language are extraordinarily complex, in general. However, it is possible to develop a procedure for *prototypical proofs* that is relatively simple. If concepts are represented by prototypes and deformations, then complex disjunctive proofs can be replaced by proofs in a prototypical model and correctness can be checked simply by applying the transformations. Coherence is then expli-
4 Conclusion

Ten years have passed since the conference in Swansea, Wales, and the small group that met in Clyne Castle under the stewardship of Bryan Niblett has grown into a large international community. What is the status of AI and Law today? In this paper, I have given a personal answer to this question.

Two distinct areas of research were surveyed: the practical work on intelligent legal information systems, and the theoretical work on computational models of legal reasoning. In each case, though, I have identified the knowledge representation problem as the most serious issue facing the field. In order to move beyond the current generation of rule-based expert systems, we need to solve the knowledge representation problem. In order to develop more adequate theories of legal reasoning, we need to solve the knowledge representation problem. My own approach to these issues is embodied in my Language for Legal Discourse [34], but there are other alternatives [25, 18].

I have omitted one recent trend in artificial intelligence from this survey: connectionism [49]. In the AI and Law field, only Rik Belew and Daniel Rose have seriously pursued a connectionist approach [2, 48], and their work has...
been confined to the practical problem of conceptual legal retrieval. On the theoretical side, Michael Dyer, who has contributed to the field of AI and Law in the past [17], has more recently become a convert to connectionism [13]. So far, he has not applied the new connectionist approaches to the law. However, one of Dyer’s main interests is the relationship between “subsymbolic” connectionist models and the more traditional “symbolic” AI models, and this is an issue of great importance for our theories of legal reasoning. Charles Walter has also emphasized this point [61]. I believe that the theory of prototypes and deformations outlined in Section 3 is an appropriate vehicle for the amalgamation of these two schools of thought. But that is yet another story.

References


REFERENCES


REFERENCES


REFERENCES


REFERENCES


REFERENCES


REFERENCES


