Some Requirements on an Action Language for Legal Discourse

(Position Paper)

L. Thorne McCarty
Computer Science Department
Rutgers University
New Brunswick, NJ 08903, USA
mccarty@cs.rutgers.edu

My interest in theories of action arises from my work on AI and Legal Reasoning. This work dates back a number of years [McCarty, 1977], and it includes both a practical and a theoretical component.

On the practical side, the goal is (eventually) to construct an “intelligent legal information system” [McCarty, 1983a]. Such a system would include at least the following modules:

1. A Legal Analysis System. This is what most people have in mind when they talk about legal expert systems. Given a representation of the rules and regulations in effect in a particular area of the law, the system would provide an analysis of the legal consequences of a given set of facts. There are a number of such systems in existence today, e.g. [Michaelsen, 1984; Susskind and Capper, 1988], but they are based on very primitive (rule-based) technology.

2. A Legal Planning System. Clients are best served, not by litigation after the fact, but by careful planning beforehand to avoid adverse legal consequences. Think of this (abstractly) as a search through a space of possible transaction patterns to identify a course of action which, according to the rules of the legal analysis system, best achieves some given set of goals. For a discussion of how we might automate this search in one important legal area – estate planning – see [Schlobohm and McCarty, 1989].

3. A Conceptual Legal Retrieval System. Legal retrieval today is based on full-text keyword search systems, using technology that has not changed substantially since the 1950’s. As an alternative, various researchers have proposed the development of “conceptual” legal retrieval systems [Hafner, 1981], which could be integrated with the legal analysis and planning systems mentioned above.

What are the obstacles to building such an integrated system? I have argued for several years that the main obstacle is the knowledge representation problem [McCarty, 1990]. So far, all attempts to advance beyond the most primitive rule-based expert systems in law have floundered on the difficulties of representing the “common sense categories” that appear everywhere in legal discourse. Accordingly, one goal of my current work is to contribute to the solution of this problem, and to establish a better technical foundation for the legal information systems of the future.

More challenging still is a theoretical goal: We would like to understand legal reasoning and legal argument in computational terms. In earlier work [McCarty and Sridharan, 1981], I proposed a theory in which the argument for a particular legal result was explained in terms of a prototypical case plus a carefully chosen sequence of transformations of this case into certain less prototypical exemplars. Again, despite some attractive properties, the theory floundered on the knowledge representation problem. (This, of course, was also the fate of most of the ambitious AI theories of the early 1980’s.) To see the problem, 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 important semantic relationships in the domain in which it is applied (here: corporate law and corporate tax law). However, the AI languages of the early 1980’s were very poor in representing common sense semantic categories. My response was to rework the semantic foundations, developing what I now call a Language for Legal Discourse (LLD) [McCarty, 1989]. I am currently engaged in a new effort to implement the earlier theory – using LLD as the representation language – and the initial results are encouraging. In addition, although the specification of LLD was driven by a crisis in theory, I believe it will also provide a foundation for the development of practical legal information systems, as discussed above.

What does this have to do with actions? My Language for Legal Discourse includes a number of diverse features: it is an order-sorted logic, with count terms and mass terms, several kinds of negation, several modes of definition and a logic programming style of computation. But it is also fundamentally a language of time and ac-
tion, both of which are ubiquitous in legal domains. In particular, I have found the following features of the action language to be absolutely essential:

1. **Indefinite Temporal References.** Explicit times and dates are certainly necessary (e.g., March 27, 1994), but we also need to refer to times that are known only by their relative positions in a temporal order. Often these references are to temporal intervals, which may be overlapping and which may have indefinite end points. Thus the expressive power of Allen’s temporal logic [Allen, 1984] seems necessary.

2. **Multiple Agents and Concurrent Events.** If the world had only a single agent, a legal system would not be necessary. Thus the representation of multiple agents in a legal knowledge base is the norm, rather than the exception. Furthermore, the actions of these agents cannot be constrained to a sequential order, by such artificial devices as interleaving. Full concurrency is necessary.

3. **Complex Actions.** Most legal categories consist of complex actions defined in terms of more primitive actions. My earliest work on corporate tax law [McCarty, 1977] includes many examples: the definition of a “stock distribution”; the definition of a “Section 368(a)(1)(B) reorganization”; etc. A minimal requirement for a knowledge representation language in a legal domain is the ability to represent these kinds of definitions.

4. **Repetitive Actions.** Many of the complex actions in a legal domain are repetitive actions. That is, if A is itself an action, either primitive or complex, we often need to define the action “doing A some finite number of times”. An example, chosen again from corporate tax law, is the concept of a “creeping acquisition”. The tax code attaches certain legal consequences to the “acquisition . . . of control”, which is defined as the purchase of a certain amount of voting stock in the acquired corporation. But this stock can be purchased in one step, or two steps, or any finite number of steps, over an indefinite interval of time, and the precise manner in which it is purchased may itself have legal consequences. We clearly need to be able to represent – and reason about – such actions.

5. **Modalities Over Actions.** The most familiar modalities over actions are deontic: permitted, forbidden, obligatory. Since these modalities appear frequently in legal domains, I have devoted some effort to the study of their semantic and computational properties, dating back at least a decade [McCarty, 1983b]. Often, the mere statement of the facts of a case requires the use of deontic modalities, as in the description of the “trust” at issue in many estate planning cases [Schlobohm and McCarty, 1989]. This requirement for the representation of deontic concepts in a legal knowledge base is now widely recognized [Jones and Sergot, 1992].

6. **Natural Language Semantics.** Finally, it is important to note that the language of legal reasoning and legal argument is natural language: English, German, French, etc. This does not mean that an intelligent legal information system must be able to cope with natural language text (although a restricted natural language interface would be very useful). Rather, it means that whenever we reach a choice point in our design of a knowledge representation language, we should opt for those features that seem to be most congruent with the underlying natural language semantics. I have found this to be a reliable methodological guide, often pointing the way to distinctions that turned out later to be justifiable on more pragmatic grounds. One example is the representation of “free choice permission” in my work on the deontic modalities [McCarty, 1983b].

For the technical details of the action language used in LLD, the reader should consult my papers in KR’92 [McCarty and van der Meyden, 1992] and KR’94 [McCarty, 1994]. The main idea in [McCarty and van der Meyden, 1992] is to treat actions as predicates over a linear temporal order, and to draw a distinction between basic actions and defined actions. Defined actions are represented by Horn clauses with (optional) linear order constraints, and the defined predicates are circumscribed to capture the intended interpretation of the Horn clauses as definitions. With this device, it is easy to represent concurrent events and repetitive actions. It is also easy to select an appropriate temporal ontology for a particular application, since discrete time and continuous time are just minor variations of a single logical language. These ideas are extended in [McCarty, 1994] to include a model theory for the deontic concepts: permitted, forbidden, obligatory. Two subsequent papers, not yet published, will present a (partial) deontic proof theory, and an analysis of negation-as-failure as applied to the deontic modalities. The techniques for answering queries in all of these languages are based on ideas originally presented in [McCarty and van der Meyden, 1991].

This paper has argued that a *Language for Legal Discourse* requires an action language with certain expressive features. I have not, so far, made any overt claims about application areas outside of the law. However, there may be other applications with similar properties: e.g., an information system in a bureaucratic environment, or an automated system for commercial transactions. (We might call these “quasi-legal” applications.) Somewhat further afield is Shoham’s proposal for “Agent Oriented Programming” [Shoham, 1992]. Could these
application areas benefit from a study of the legal domain? Would LLD be useful here as well? These might be interesting questions for discussion at the AAAI Symposium.

References


