1-1-2001

Legal Reasoning and Artificial Intelligence: How Computers "Think" Like Lawyers

Kevin Ashley

Karl Branting

Howard Margolis

Cass R. Sunstein

Follow this and additional works at: http://chicagounbound.uchicago.edu/roundtable

Legal Reasoning and Artificial Intelligence: How Computers "Think" Like Lawyers

Kevin Ashley, Karl Branting, Howard Margolis, Cass R. Sunstein

HOWARD MARGOLIS: I look forward to what I'm going to learn this afternoon [November 3, 2000]. Artificial intelligence and the law has its roots about twenty years ago. It has been going rather strong for the last decade, in particular in the hands of two of our guests.

Karl Branting, who is a professor at the University of Wyoming, is going to speak on some of the philosophical and broader issues. Kevin Ashley has been much more concerned with the practical problems of creating modules of encapsulated legal judgments that actually work. Cass Sunstein, who as you all know knows everything, will comment. [*laughter*]

So we will proceed in a logical fashion. Kevin is going to take twenty minutes because we want to get some concrete examples on the table so we really know what we are talking about. He will take twenty minutes to talk about three concrete examples. Then, Karl will talk in a more philosophical way about how research such as Kevin's links to other things going on in the area of artificial intelligence and scientific discovery and how well it's doing within law. Cass, then, will comment on whatever they say. I am allowed to say as few declarative sentences as possible and to ask questions. [*laughter*] And so we can proceed, beginning with Kevin Ashley of the University of Pittsburgh School of Law. [*applause*]

KEVIN ASHLEY: Thank you very much, it's a pleasure to be here. I'm pretty sure I could not get a roomful of people on a Friday afternoon at the University of Pittsburgh Law School to discuss AI[1] and law, but the free booze would help.

I want to provide three examples of computational models of legal reason-

---

1. Artificial intelligence.

ing and show what you can do with them. Computational models of analytical legal reasoning are comprised of a knowledge representation and an inference mechanism. The knowledge representation captures some important aspects of legal knowledge. And the inference mechanisms are algorithms that enable a program to use those elements of legal knowledge that are represented in order to solve problems.

Slide 1
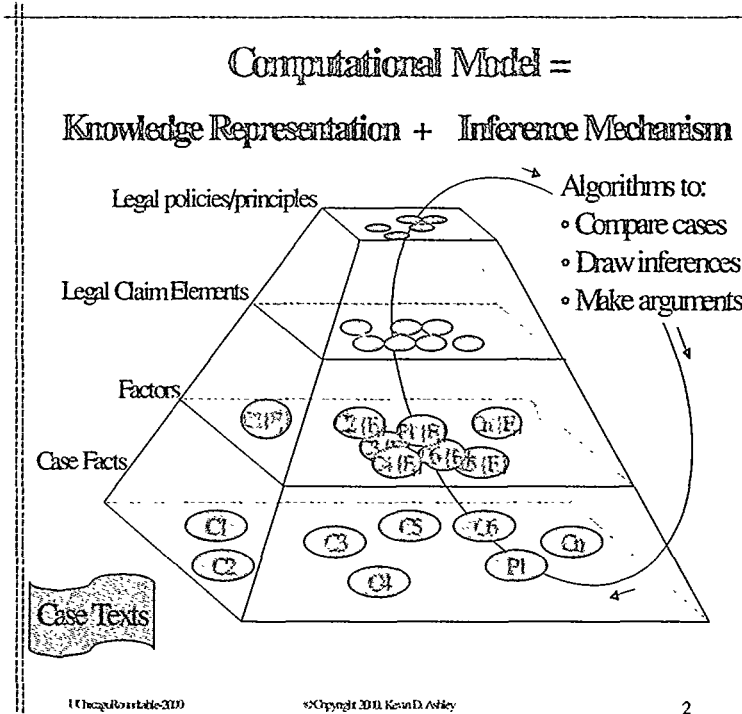
This is an idealized illustration of a computational model of legal reasoning. It comprises a knowledge representation and an inference mechanism. The knowledge representation is a conceptual hierarchy of legal information dealing with some particular type of legal claim. At the bottom level it relates cases and their facts to the elements of some legal claim and, ideally, ultimately to the legal policies and principles that underlie that legal claim. The algorithms of the inference mechanism use that information. For instance, the inference mechanism may take a problem situation and compare it to other cases in light of the other cases' analyses, draw inferences about how that problem should be decided, and generate arguments using the information in the computational model.

Now there are three general issues in designing these computational models.

One is: How does one connect the facts of cases to the statutory elements of a legal claim? My approach in two programs that I have worked on, HYPO[2] and CATO,[3] was to introduce an intermediate level of factors. These are stereotypical patterns of facts that tend to strengthen or weaken a plaintiff's argument in support of its claim. In my work, when the program compares a problem to cases, it is comparing them in terms of these factors. In Karl Branting's GREBE program,[4] a different approach was used.

Secondly, notice that the case texts are not in the model, not in the knowledge representation. This is because AI programs can't read yet. They can't understand natural language text in general. So someone has to represent the facts of the case manually in such a way that the program can know what the facts are and can determine how to analyze the facts.

The third general problem is how does one implement, how does one represent, the underlying legal principles and policies of a legal claim? How does one implement their roles in analyzing cases? And what about the dialectical role of the cases in filling out the meanings of these abstract legal principles and policies? This is a problem that I have not solved in any program, but I have some ideas and I hope that I can show you a couple of them.

---

2.   HYPO is a program that performs case-based legal reasoning in the domain of trade secret misappropriation law. See Kevin Ashley, *Modeling Legal Argument: Reasoning with Cases and Hypotheticals* (MIT 1990).

3.   CATO (Case Argument TutOrial) is described in Vincent Aleven, *Teaching Case-Based Argumentation Through a Model and Examples* (1997) (PhD dissertation, University of Pittsburgh Graduate Program in Intelligent Systems), available at http://www.cs.cmu.edu/~aleven/dissertation.html. See also Kevin D. Ashley, *Designing Electronic Casebooks That Talk Back: The CATO Program*, 40 Jurimetrics J 275 (2000).

4.   GREBE (Generator of Recursive Exemplar-Based Explanations) is a computer system that integrates legal precedents with statutory and common sense rules for legal analysis. See L. Karl Branting, *Reasoning with Rules and Precedents: A Computational Model of Legal Analysis* (Kluwer 1999); L. Karl Branting, *Building Explanations with Rules and Structured Cases*, 34 Intl J Man-Machine Studies 1 (1991); L. Karl Branting and Bruce W. Porter, *Rules and Precedents as Complementary Warrants*, Proceedings of the Ninth National Conference on Artificial Intelligence (AAAI-91), Anaheim, California, July 14-19, 1991; L. Karl Branting, *Reasoning with Portions of Precedents*, Proceedings of the Third International Conference on Artificial Intelligence and Law, Oxford, England, June 25-28, 1991.

Slide 2[5]
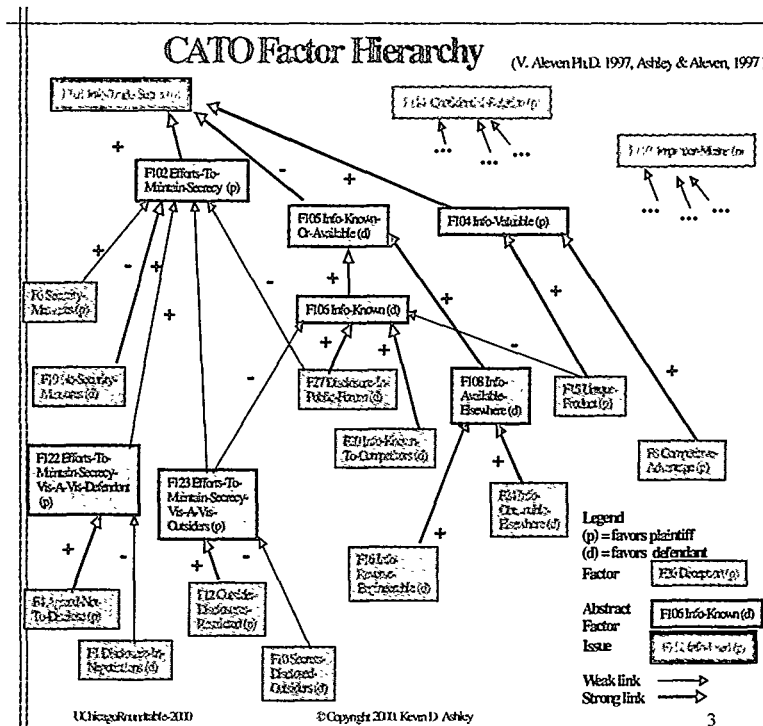
Now that's an idealized model. This is about as close as I've come to realizing that type of computational model. This is a Factor Hierarchy that my student, Vincent Aleven, designed for the claim of trade secret misappropriation. At the top level are the elements of the claim. For example, is the information a trade secret? Is there a confidential relationship? Were improper means used? And for each of those there would be a Factor Hierarchy. I'm just showing you the Factor Hierarchy for one: Is the information a trade secret?

At the bottom level are the various factors, the stereotypical patterns of facts that strengthen or weaken a claim. For instance, here's F15, factor 15: "unique product." It stands for a stereotypical fact pattern that one often sees in trade secret cases. The plaintiff's claim is stronger to the extent that its product is unique in the industry. It's relevant to the issue of whether that information is a trade secret in two different ways. It shows that the information is valuable. It also suggests that the information is not known in the industry. The Factor Hierarchy is a graph, that is to say any given factor on it can have more than one

---

5. Adapted from Aleven, *Teaching Case-Based Argumentation Through a Model and Examples* Figure 3-2 (cited in note 3).

parent and we'll come back to the significance of that in just a second.

Slide 3

Factor Hierarchy ⇒ Multiple Interpretations

▶ Where pltf. and def. entered into a nondisclosure agreement [F4], pltf. took measures to keep its information secret [F6], and def. entered into an agreement not to compete with pltf. [F13], pltf. should win a claim of trade secrets misappropriation, as in the *Elcor Case.*

◀ *Elcor* is distinguishable. It is stronger for pltf. than is the current problem. In *Elcor*, pltf.'s product was unique on the market [F15] and there were substantial similarities between pltf.'s and def.'s products [F18]. Not so in MBL. ...

▶ [F15] is not an important distinction. In MBL, def.'s access to pl.'s info enabled it to develop its product in less time or at lower cost [F8]. It follows that in both cases, pl.'s info was valuable for pl.'s business [F104].

◀ [F15] is a marked distinction. Shows that in *Elcor*, the info apparently was not known outside pl.'s business [F106], whereas in MBL, pl.'s info was known outside pl.'s business [F106]: Pl. disclosed its product info to outsiders [F10] and pl.'s info was generally known in the industry [F20].

UChicagoRoundtable-3/00   © Copyright 2000, Kevin D. Ashley   4

One can use a computational model like this to generate legal arguments and even to generate alternative interpretations of cases. This is a sample argument that the CATO program generates for a problem called the MBL case.[6] I don't want to get into the details here, but I just want you to see the structure of it. It starts with an argument by analogy (first row of Slide 3). CATO argues that the MBL case should have the same result as the Elcor case,[7] that is, the plaintiff should win. And it elaborates an analogy in terms of the factors that the two cases share. Then the program switches hats and argues that Elcor is distinguishable from the MBL case (second row of Slide 3). It's now arguing on behalf of the defendant. It points out, among other things, that the product was unique in Elcor, that is to say that factor F15 applies. But that was a strength that one doesn't find in the MBL problem.

I would also like you to focus on the last two parts. Here, CATO downplays

---

6. *MBL (USA) Corp v Diekman*, 445 NE2d 418 (Ill App 1983).
7. *Elcor Chem Corp v Agri-Sul, Inc*, 494 SW2d 204 (Tex Civ App 1973).

the significance of that distinction on behalf of the plaintiff (third row of Slide 3). It argues that the fact that the product was unique in Elcor is not an important distinction. It argues that the reason that that factor matters is that it shows that the product is valuable, that it has value. That was abstract factor 104 in the Factor Hierarchy. CATO points out that in the MBL case, there was other evidence that the product was valuable. In other words, CATO is arguing, given the reason why the distinction matters, that these two cases are basically the same. In this last argument, CATO responds for the defendant, it switches hats again, and now it's emphasizing the significance of this distinction (fourth row of Slide 3). It's saying here that the reason that that factor matters is that it shows that the information was not known outside of the plaintiff's business. And it goes on to say that in MBL there's evidence that the information was generally known, and it points to some other factors in the MBL case. In other words, CATO is arguing that, given the real reason why the distinction matters, the two cases are really quite different. In other words, it's doing analogical reasoning here.

In downplaying and emphasizing this distinction, from where does the knowledge come about why these differences matter from a legal point of view? Well, it comes from the Factor Hierarchy (Slide 2). Let me just show you that again. In making this argument, downplaying and then emphasizing the distinction, CATO is working up from factor F15 on two alternative paths through that Factor Hierarchy. In arguing for the plaintiff that the cases are similar, it's drawing an analogy at the level of that abstract factor 104 that this information is valuable. In arguing that the cases are actually quite different, it's following a different path from F15 through 106 and making a connection to other facts in the MBL case, factor F10 and F20, and using that information to argue at an abstract level that these cases are really quite different. Now, CATO has algorithms that enable it to decide which paths to follow in this Factor Hierarchy and how high up to go in selecting an abstract way of characterizing the significance of the differences. Those are the algorithms. That's the inference mechanism.

We use this feature in a program that teaches law students basic argumentation skills. And my student Vincent Aleven, in his dissertation,[8] evaluated the CATO program in a controlled experiment involving first-year legal writing students with some good results. Another practical application of it might be as a kind of brief writer's assistant. One could imagine having a specialized Factor Hierarchy and case database that's updated periodically with good coverage of a particular kind of claim and that would help associates in a law firm analyze claims and make arguments. So that's my first example of a computational model.

My student, Steffi Brüninghaus, is using a similar kind of computational

---

8.   Aleven, *Teaching Case-Based Argumentation Through a Model and Examples* (cited in note 3).

model in a different way. This time it's an attempt to harness the model to help another computer program called SMILE[9] to learn to classify new texts automatically. This SMILE program learns how to assign factors to the raw text of new trade secret cases based on a corpus of manually marked-up texts that we've prepared for the CATO program. So this time, we're trying to bring those case texts into the computational model.

Slide 4
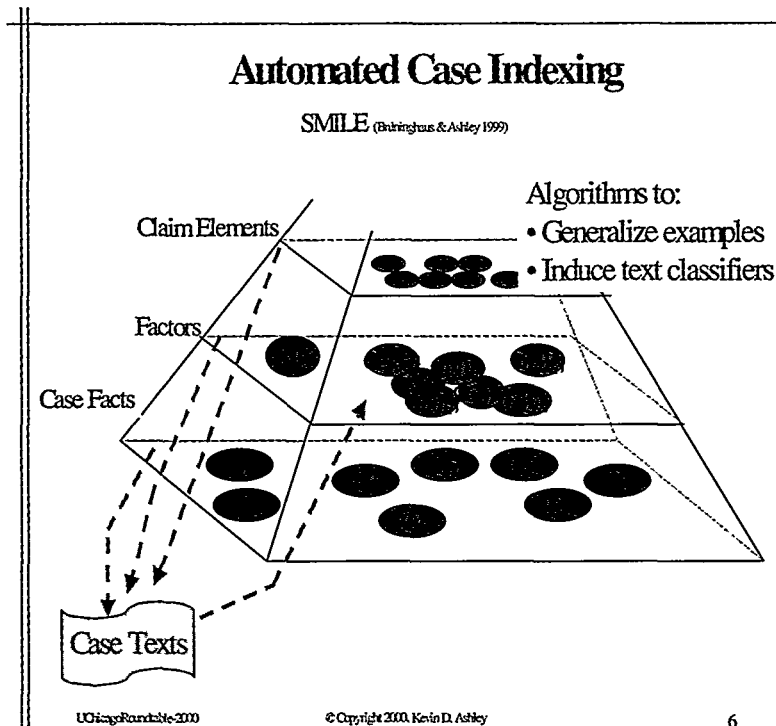
SMILE learns from a set of training instances, sentences that are positive or negative instances of a factor in a given case. The four sentences [on the left side of Slide 5] are positive instances of a factor that you've seen before, factor F15: "unique product." It's on the basis of sentences like these that a human reader might conclude that a particular factor applies to the case from which the sentences come. All the other sentences in the case are treated as negative instances of F15.

---

9. SMILE (SMart Index LEarner). See Stefanie Brüninghaus and Kevin D. Ashley, *Toward Adding Knowledge to Learning Algorithms for Indexing Legal Cases*, in Proceedings of the Seventh International Conference on Artificial Intelligence and Law (ACM 1999), available at http://www.pitt.edu/~steffi/papers/icail99.ps.

SMILE uses a learning algorithm called ID3[10] to learn a decision tree for distinguishing the positive instances of sentences from the negative instances of a particular factor. On the right [of Slide 5] is a part of the decision tree that SMILE actually learned from the positive instances. It corresponds to a rule for classifying sentences. If the text of the sentence includes the term "unique," then conclude that factor F15 applies. Otherwise, if it contains "known" and "general," conclude factor F15 applies, and so forth. It may seem like a naïve rule, but in an evaluation we showed that it does a pretty good job of distinguishing the sentences that are positive instances of a factor from those that are not.

Slide 5
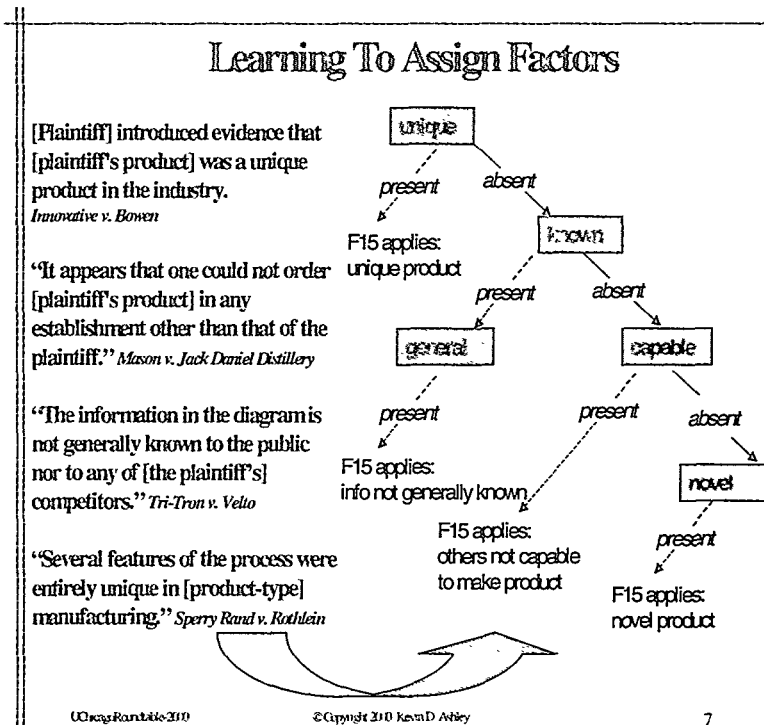
To refine the decision trees we need to get more mileage out of the examples. Steffi has focused on trying to make the examples more general. For instance, take a look at that first sentence [in Slide 5]. It actually said in the original "Innovative." That's the name of the plaintiff. "Innovative introduced evidence that Panl Brick," that's the name of the plaintiff's product, "was a unique prod-

---

10. See Ross Quinlan, *C4.5: Programs for Machine Learning* (Morgan Kaufmann 1993).

uct in the industry." Obviously, the terms "Innovative" and "Panl Brick" don't appear very often in trade secret cases. I made the substitutions of "plaintiff" and "plaintiff's product" manually to make the examples more general. If we can get SMILE, as a program, to make these kinds of substitutions automatically, then it can generalize the instances itself and learn more powerful rules from them. For instance, a rule might be: "If plaintiff's product is unique, then conclude factor F15." We're using information extraction to create better examples for the program to generate more powerful rules.

Where would this be useful? Well, for one thing, if you've got a program like CATO, SMILE would automatically add new cases to the database. For another, if you are using Westlaw, you know that Westlaw retrieves texts and orders them by statistical criteria. A program like SMILE could process those cases that are retrieved and highlight the important stereotypical facts and weaknesses in the text of the case.

Okay. Now I'm down to my last example. And the final example is a computational model for practical ethical, rather than legal, reasoning. My former student Bruce McLaren studied a set of more than four hundred decisions of the Board of Ethical Review of the National Society of Professional Engineers. He created a program called SIROCCO[11] which, given a problem situation, retrieves past ethics cases and ethics code provisions that are relevant to the analysis of the problem. We used that program to investigate empirically an interesting feature of ethical reasoning that also applies to law.

Normative principles in professional ethics are very abstract rules. For instance, here's an example: "Engineers shall recognize that their primary obligation is to protect the safety, health, and welfare of the public." Now, how does one know how to apply that abstract principle? In ethics, as in law, principles cannot be defined intentionally and applied deductively. There are no readily available sources of authoritative rules that bridge the gap between the high-level principles of the top and the low-level factual scenarios at the bottom. Nevertheless, we observed that, like judges, the Board, in deciding cases, cited relevant ethics code provisions and also cited relevant past cases. And we hypothesized that the decided cases and their explanations of how the principles apply, in effect, flesh out a meaning for those very abstract principles at the top level. We say that the cases "operationalize" the principles. We believed that these case-based extensional definitions of the principles could be represented and also used for improving the program's retrieval ability. Now, notice that this time I have not used factors as a way of bridging the gap between the case facts and the higher level principles. Instead, I've used something called instantiations,

---

11. SIROCCO (System for Intelligent Retrieval of Operationalized Cases and COdes). See Bruce McLaren, *Assessing the Relevance of Cases and Principles Using Operationalization Techniques* (1999) (PhD dissertation, University of Pittsburgh Graduate Program in Intelligent Systems), available at http://www.pitt.edu/~bmclaren/publications.html.

which are the result of the Board's exercise of these operationalization techniques, in effect, to bridge the gap between case facts and abstract principles.

Slide 6
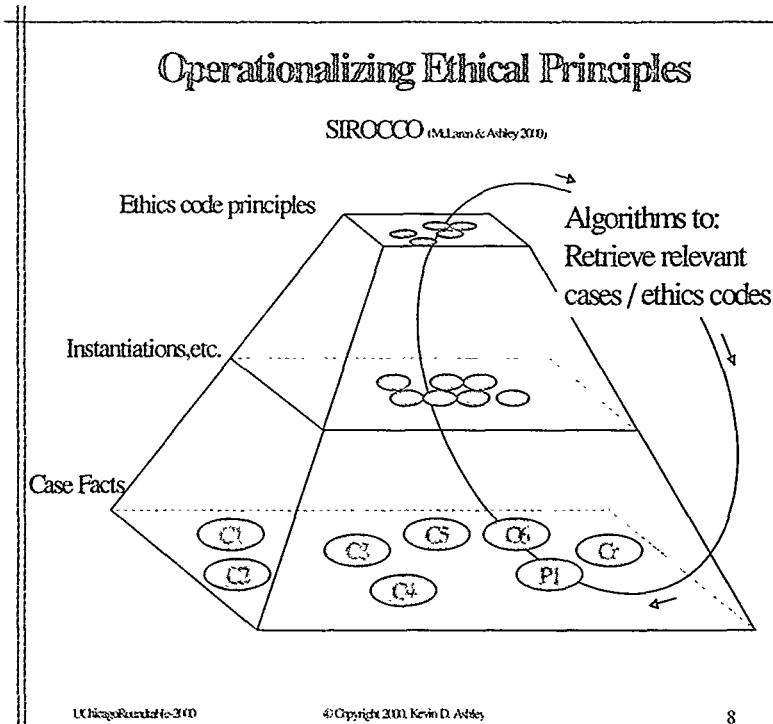
Let me show you a little about how operationalizations link facts and abstract principles. [Slide 7] contains a list of the operationalizations that Bruce cataloged during his analysis. These are the Board's techniques for bridging the gap between the abstract codes and the specific fact situations. For instance, code instantiation is a kind of more concrete interpretation of a code in the context of a case. In essence, the Board links combinations of selected facts in the case to a code provision when they instantiate it. For instance, in one case, the Board instantiates that public safety code provision that I cited to you before. ["Engineers shall recognize that their primary obligation is to protect the safety, health, and welfare of the public."] It was a case involving a building inspection. Basically, the Board said that once an engineer discovers that an apartment building, which he has been hired to inspect, presents a danger and he knows that the building authority should be informed, then he has an obligation to warn of the danger even though his client instructs him to withhold the infor-

mation. Well, that's a little concrete set of facts with which we annotate the broad principle and then we can use that information in improving the program's retrieval ability. I'll just close by showing you how that works.

Slide 7

Operationalization Techniques

Code Operationalizations in NSPE BER Case Set:

(1) *Code Instantiation*

(e.g. of II.1.a. from 90-5-1: "...It appears that Engineer A, having become aware of the imminent danger to the structure, had an obligation to make absolutely certain that the tenants and public authorities were made immediately aware of the dangers that existed.")

(2) *Apply Code to Hypothetical Scenario*

(3) *Rewrite a Code*

(4) *Group Codes*

(5) *Designate Superior Code in Context*

(e.g., II.1.a. over II.1.c. from 90-5-1: "...in cases where the public health and safety is endangered, engineers not only have the right but also the ethical responsibility to reveal such facts to the proper persons.")

UChicagoRoundtable-2.00   ©Copyright 2001. Kevin D. Ashley   9

We compared SIROCCO to five other methods, including an oblated version of SIROCCO, that is to say, a version of SIROCCO that doesn't use the operationalization knowledge. We called that "Non-Op SIROCCO."[12] (This is a nice feature of a computational model. One can turn a knowledge source off or on at will for purposes of experimentation.) We performed the tests using a case base of 184 foundational cases and fifty-eight trial or test cases. Historically, the fifty-eight test cases came after the foundational cases. For each trial case, we compared the cases and code provisions that SIROCCO, the program, recommended as being relevant with the cases and code provisions that the Board actually said were relevant. And we compared the overlap using a number called the F-measure. It's a combination of precision and recall. The graph [in Slide 8]

---

12. See McLaren, *Assessing the Relevance of Cases and Principles Using Operationalization Techniques* at 21 (cited in note 11).

shows the mean F-measure per method over all fifty-eight trial cases. We used two different kinds of experiments here.

The point I want to make to you now is that in both instances SIROCCO outperformed Non-Op SIROCCO and the differences were statistically significant. This difference is the contribution to retrieval effectiveness that comes from the Board's operationalizations. We were able to sort of bottle it, if you will, and take advantage of it in improving the program's ability to retrieve cases.

Slide 8[13]

UChicagoRoundtable-2010   ©Copyright 2010 Kevin D. Ashley   10
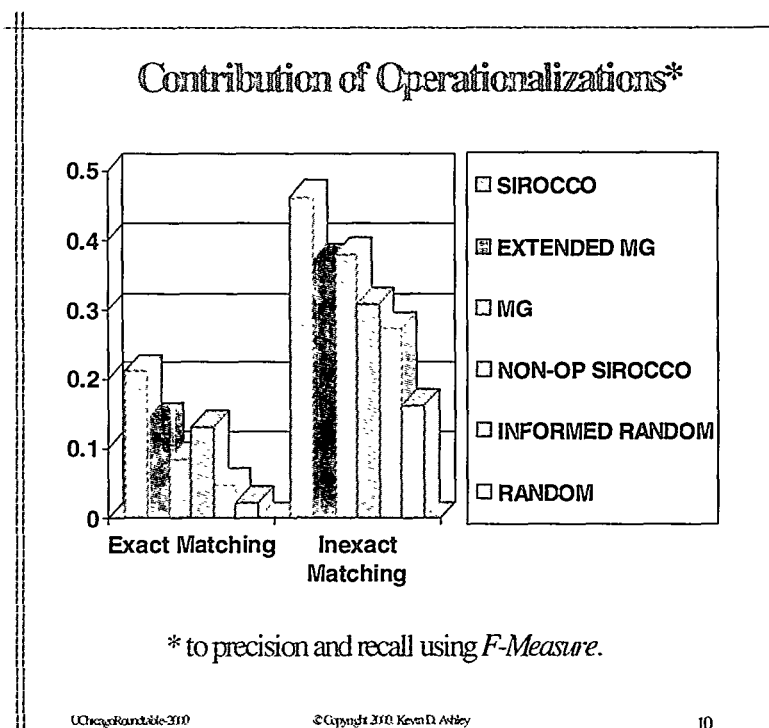
So, in conclusion, those are my three examples of computational models and what you can do with them. You can generate arguments with them and even generate alternative interpretations of the significance of important similarities or differences. You can use them to help a program learn to index cases automatically. And you can use them to conduct interesting empirical investigations of such phenomena as operationalizing principles.

Thank you. [*applause*]

13. Adapted from McLaren, *Assessing the Relevance of Cases and Principles Using Operationalization Techniques* Figure 4-5 (cited in note 11).

MARGOLIS: And we turn next to Karl Branting.

KARL BRANTING: Thank you, Howard.

So Kevin has presented several computational models of legal reasoning and how they can be used for analysis and for tutoring. It's my opinion that these models can be useful in jurisprudence for helping to evaluate alternative jurisprudential theories by actually implementing them and testing on examples. But rather than pursuing that question, what I'd like to talk to you about, in the few minutes that I have for my presentation, is to try to make the case that this field is relevant to all of you. I think that, in the long term, it is going to change the character of the American legal system. I am going to make my case slightly more emphatically than I really believe it, just to engender a little bit of discussion about this.

But anyway, my claim is that the development of computational models of legal reasoning that can actually be used for problem solving—systems that I'll call legal expert systems—are really going to change the practice of law and the American legal system, and that this is going to happen during your careers. To substantiate this claim, I am going to appeal to five separate factors, which I'll enumerate, and then I'll say a little bit about each one of them.

Here's factor number one, the one I'll talk about the most, the claim that legal expert systems, adequate for many routine legal problems, already exist and new computational models will continue to be developed. That's claim number one. Claim number two is that there is a vast, unmet demand for legal services by the public. Richard Susskind, in his book *The Future of Law*,[14] terms this the latent market—all those people out there who can't afford lawyers. Three, the World Wide Web constitutes a kind of electronic infrastructure for the distribution of legal services. Four, legal expert systems constitute a new vehicle for marketing and distributing the expertise of lawyers. And five, funding limitations on governmental bodies, which we know are always short of money, are inevitably going to drive them to automate a larger and larger proportion of the services that they deliver.

So, in summary, I am suggesting that legal services can be viewed as a kind of commodity, for which the public is the consumer, attorneys are the producers, the World Wide Web is the highway, and legal expert systems are the vehicles. And by decreasing distribution costs and increasing economies of scale, legal expert systems will inevitably lead to increased consumption of this commodity. Less efficient producers will inevitably be priced out of the market. So that's my claim.

Let me consider each of these factors in turn. The one that I'll talk about the most is number one, the claim that legal expert systems adequate for many routine legal tasks already exist. To substantiate this claim actually would take quite a bit of work, maybe a whole semester class would get it started, so I am

---

14. Richard Susskind, *The Future of Law* (Oxford 2d ed 1998).

only going to be able to make a few comments in support of it.

Let me start by making the observation that there are quite a few different participants in the legal system and, in my view, for each one of these participants there may be various different legal tasks each requiring a separate computational model. By the participants, I mean we have the members of the public, clients, attorneys, judges, clerks, legislators, all people who are participants in the overall process. Just for simplicity, let's hone in maybe on the most typical garden variety sort of legal problem solving episode that we might imagine, which is when an individual comes to consult an attorney about some legal problem.

What kinds of separate tasks does an attorney perform? Well, I claim that there is a whole series of them. The first one is what I call problem formulation. When a client explains a problem to an attorney, the attorney has to elicit the legally relevant facts, steer the client away from the legally irrelevant stuff (indignation and so forth), and the attorney needs to formulate the problem that is being posed by the client in terms of legally relevant concepts. The second step is retrieval. The attorney, the legal problem solver, needs to think of some legal authorities that are relevant to the problem that has been formulated in the first step. Next is what we might call problem analysis. This is determining what sort of legal consequences might follow from the application of the legal authorities to the facts as elicited by the attorney. Next is the task of prediction. That is, for each of the possible outcomes of some legal action, litigation for example, estimating the probability of those outcomes. What is the expected return on them? How much would it cost to go to trial on a certain issue? How much might you win? How likely are you to win? Other tasks are planning, deciding what sort of actions should be taken on behalf of the client's goals, document drafting, and others. I won't enumerate them all.

The character of each of these tasks is rather different from the others. Not all of these tasks are amenable to modeling, only a subset. But the subset that, in my view, is most amenable includes the task of retrieving authorities, the analysis task, the prediction task, and the document drafting task. I'll just hone in on those. So far, I am still trying to support my contention that there are models of legal reasoning that are adequate for routine tasks. Number one, retrieval, I'm not going to talk about. There is a long history of AI models doing it. But right now the dominant models, e.g., the LEXIS/Westlaw kind of model, don't use much in the way of AI. I won't talk about that one. Instead I'll start with analysis.

Analysis. That's the idea that if I have some well-defined facts and I have some well-defined authorities, can I derive some arguments from them? Well, I think that if the problem is sufficiently well posed, then there are a variety of different models for finding arguments for and against a different legal conclusion. The simplest and historically the oldest one simply maps legal rules onto computational rules or logical rules. Such systems are useful if the case facts are

relatively stereotyped and clear-cut and it's the legal rules themselves that give rise to the complexity of case analysis, rather than the vagueness, ambiguity, or context dependency of the legal concepts in those rules. So, in other words, something like the UCC[15] is much more appropriate than a legal problem involving reasonable care, let's say.

Computational models based on these rules were first developed in the 1970s. They are nothing new. They, in turn, were based on logical models developed notably by Layman Allen in the 1950s.[16] So they have been studied for quite a long time. People have also been familiar with the weaknesses of these models for quite a long time, because we know that lawyers don't treat legal rules as a static body of legal formulations but rather legal rules are the tools that lawyers use for achieving their goals. Now the kinds of models that Kevin has been showing you are more sophisticated. They involve reasoning by analogy. As it happens, the earliest analogical models were developed in the 1970s and the early 1980s. These analogical models tend to involve a much larger knowledge acquisition effort. In other words, constructing these systems tends to be a more involved process.

The second task that has been intensively studied is prediction. As I mentioned, after the analysis, if we are thinking of our interaction between a client and an attorney, there is the legal analysis, but then prediction is an important part as well. Now the largest consumers of predictive systems are insurance companies, which formalize the expertise of claims adjusters and attorneys through a lengthy process of interview and observation to produce systems that predict the settlement value of insurance claims. So there are a large number of such systems, but they are almost always proprietary. Oddly enough, there is not a large amount of literature on predictive systems.

But in my view, there is reason to believe that all of the participants in the legal system could profit from predictive systems. Psychological studies, notably by Elizabeth E. Loftus and W. Wagenaar,[17] have shown that attorneys systematically overestimate their likelihood of success at trial. [*laughter*] Why is that? Well, there is a reason for it. Optimism is rewarded. In fact, the most successful trial lawyers are those whose estimates are least realistic, that is, are most overly optimistic. So what does this mean? This means that as an institution, courts are rewarding behavior that isn't optimally beneficial to the system as a whole. In other words, the best strategy for an attorney is not necessarily the best for the client. And it is almost certainly not best for a society as a whole that has to pay for lawsuits that would never take place if people had a realistic estimate of their

15. Uniform Commercial Code.

16. Layman Allen, *Symbolic Logic: A Razor-Edged Tool for Drafting and Interpreting Legal Documents*, 66 Yale L J 833 (1957).

17. Elizabeth E. Loftus and W. Wagenaar, *Lawyers' Predictions of Success*, 28 Jurimetrics J 437 (Summer 1988).

probability of success or, more precisely, the expected return on the lawsuit.

A third task is document drafting. As we all know, there is already a large commercial market for the very simplest document drafting systems. There are lots of sophisticated models of document drafting, including some based on state of the art linguistics and speech-act theory.

So in summary, I think that there is, at least for these three tasks—for document drafting, prediction, and analysis—a history of computational models. They vary widely in their flexibility and explanatory power and development costs. And moreover, the relative merits of these models are, of course, a matter of dispute among computer scientists and scholars of jurisprudence. But the fact is that, at least at the low end, executable models—legal expert systems—already exist.

So that was all on claim number one. My other claims will be briefer. Factor one, the claim that models exist. Two, the unmet need for legal services. I think I can just appeal to the familiar experience that we all have that attorney fees are quite expensive. As a result, individuals are frequently unable to afford answers to basic legal questions. The cost of getting an answer to a legal question is often greater than the value of the claim that the question applies to. Of course, that is not true of large institutions, but for ordinary citizens this tends to be the case. And these costs are exacerbated by uncertainty in the legal system, which is the result of the fact that the law is in a state of evolution and therefore unsettled, the delays of litigation, which are worse in some places than in others, and the overall lack of predictability in the process.

So far I've said there are good models. There is also an unmet demand for legal services. A third factor is the World Wide Web. The World Wide Web is a wonderful development for people in my area of study because it provides a uniform computer interface familiar to a very high proportion of litigants and attorneys. It largely eliminates the problem that we used to have in software distribution, that is, hardware inconsistencies and interface inconsistency and unfamiliarity. So even the most technology averse lawyer is likely to be familiar with web browsers, if only because he or she has seen his or her children using them. *[laughter]*

Claim number four, legal expert systems as a distribution mechanism for legal expertise. It is my observation that the economic motivation for the law firms that are most active in development of web-based legal expert systems— such as London-based Linklaters, Sidney's Blake, Dawson & Waldren, Ernst & Young, and others[18]—is that, well, first of all, that legal expert systems perform work that wouldn't otherwise be done by the firm. So the idea is that clearly you don't want to make an expert system that you market for less than the amount it would cost one of your own attorneys to perform the same work. But the motivation is that the legal expertise can be marketed to a larger number of consum-

---

18. See Alan Cohen, *Legal Advice Without the Lawyers*, New York L J (Nov 15, 1999).

ers if it is formalized as a computer program that is then delivered over the web. And the second factor is that users of the expert systems may become customers for more complicated and more lucrative personal services. People who pay less to get some advice may then have an incentive to say, "Hey, I want to know some more than this program can deliver to me." The bottom line is that these law firms are betting that legal expert systems can improve the profits that they obtain from the marketing of their legal expertise.

And finally, the last comment about government services. There is an immense demand for routine legal information from state, local, and federal governments. This has already given rise to quite a few web-based legal expert systems produced by government agencies. Right now there are legal expert systems for the Advisors on Employment Standards Administration (ESA), the Mine Safety and Health Administration (MSHA), the Occupational Safety and Health Administration (OSHA), the Pension and Welfare Benefits Administration (PWBA), and the Veterans' Employment and Training Service (VETS).[19]

Web-based delivery of legal services is also a promising strategy for addressing the needs of pro se litigants. Of course, there are actually two arguments about that. There is one view that "Geez, if you can't afford a lawyer, how can you afford a computer or how likely are you to be able to understand how to use a computer?" But it is my surmise that familiarity with computers is becoming quite ubiquitous in our society.

So I've made this claim about the growing economic importance of legal expert systems. Let me end by adding a couple of provisos. There are some countervailing factors. One of them is technical. You may have been struck during Kevin's presentation by the fact that this system that he was showing to you seemed quite elaborate. And it is true that the expertise of these systems— we can imagine them as embodying expertise—is hard to come by. It is a very laborious process to take someone's expertise and formalize it in a manner that is executable on a computer. So the development costs of legal expert systems are very high and, in my view, they are only going to come down when there are significant improvements in knowledge acquisition, that is, the process of automating the formalization of expert knowledge. And then in particular, that improved natural language processing is going to be key. But once again, Kevin's student is a typical researcher engaged in improving those techniques.

There are also institutional and cultural barriers. A really major one is time billing. Lawyers are apt to be reluctant to make use of legal expert systems to perform some of their work if they bill by the hour and such systems reduce the amount of time it takes them to solve a problem. So time-based billing is antithetical to the acceptance of these techniques. But task-based billing, on the other hand, creates an economic incentive to the automation of the more routine things that you can automate. Partnership promotion practices discourage

19. See http://www.dol.gov/elaws.

activities that are not billable. In the firms that I mentioned earlier that have invested a great deal of effort in creating legal expert systems, for internal institutional reasons there is not a lot of internal pressure for the people involved to be billing hours constantly because there is this huge upfront cost in the time of legal experts that is only amortized over the lifetime of the use of the program.

And the last one is the rather difficult institutional barrier, the ill-defined standards for the unauthorized practice of law. I think that there is going to be litigation on this subject in increasing amounts, because there are many attorneys at the low end of the food chain that are going to be directly threatened, already are threatened, by these systems, who are going to find claims of unauthorized practice of law as a way of attempting to stanch this flood.

Let me conclude with some predictions concerning the effects on the legal community. I think that legal expert systems aren't going to reduce demand for high-end legal services. In fact, I think, to the contrary, it is going to improve the delivery of high-end legal services by automating some of the more routine aspects. As an example, there is a new product that uses natural language processing to do proofreading pretty effectively. And it is marketed by saying that this is a mechanism to retain associates who would otherwise get so discouraged at being kept up late proofreading documents over and over again that they would move to some other firm.[20] Plausible or not, I don't know.

On the other hand, the providers of routine legal services are going to face increasing competition from legal expert systems. And I think, as I said, that solo practitioners are already under such pressure. It may be that a new field of legal information engineers is going to develop consisting of attorneys whose job is to organize information for electronic mass distribution. And finally, I think that interactions with low-level government functionaries will increasingly be replaced by simple web-based legal expert systems. Am I out of time?

MARGOLIS: Yes. [*laughter*]

BRANTING: Thank you. [*applause*]

CASS SUNSTEIN: This is extremely interesting material. A major question is: What can we learn about artificial intelligence and what can we learn about legal reasoning from bringing them into contact? That's what I'm going to try to say something about.

There's a weak version of the enthusiasm for artificial intelligence in law— weak meaning less ambitious—and that is that this is like really upscale LEXIS and Westlaw. It bears the same relationship to LEXIS and Westlaw as LEXIS and Westlaw bear to *Shepard's*. In this view, it's extremely helpful for lawyers, who can find a lot of cases quickly. Plug in a problem and they'll see lots of cases like it and potential similarities and differences. That seems to me a convincing claim. Kevin Ashley has demonstrated it. That's the weak version and I'm all for that.

---

20. See http://www.expertease.com.

The strong version, which both speakers actually endorsed, is that artificial intelligence as we now have it can engage in analogical reasoning or does engage in analogical reasoning. To phrase it a little more polemically than is probably fair, I'll say that's just a mistake because at the present state of the art artificial intelligence cannot engage in analogical reasoning or legal reasoning. They can't do it. And the view that they can do it, or are doing it, is based on a misunderstanding of what analogical reasoning is, one that disregards the inescapably evaluative or normative dimension to my claim that one case is "like" another case. To engage in analogical reasoning, to do it, there has to be an evaluative argument showing that this case is like that case. There has to be a principle, and at the current state of the art, artificial intelligence can't generate good principles, or principles at all. I'm hoping this will be helpful.

Suppose you have someone who's been fired by an employer, a copilot, say, for refusing to fly an airplane on the ground that it's not safe. The employer has fired the copilot, and the copilot wants his job back or wants some money. Let's suppose, to make it very simple, that we're in a jurisdiction in which one court has held that you can't fire someone for refusing to commit a crime and another court has held that you can fire someone for reporting that the bank for which he works hasn't engaged in advertising activity in low-income communities. This is a world with just three cases: the case at hand, one case the employee wins, another case the employee loses. What's to be done? This is a problem in analogical reasoning.

A going account of analogical reasoning is by Edward Levi,[21] and the title of this subsection of my talk is "Levi's Mistake." What Levi suggested was, in engaging in analogical reasoning, judges ask which case is more similar to the case at hand or which case has more similarities to the case at hand. Is it clear that that's not a very helpful way of doing analogical reasoning in our pilot case? Is the pilot case more similar to the bank case or is it more similar to the crime case? To figure that out you can count similarities. But is that what you're going to do? It's not an exercise in counting. You have to do something else, and let's make a little amendment to Levi and say you can search for relevant similarities. Now that's helpful, or at least more helpful than counting for "more." You need to find relevant similarities and HYPO, the computer program, can do that, but that's not helpful enough. To know whether a similarity is relevant, you need to figure out the principle for which the first case stands, and the first case doesn't tell you that. Is the idea in the crime case that you can't fire someone for refusing to inflict harm on third parties? If so, then our pilot maybe is going to be okay. Or is the principle instead you can't fire someone for refusing to commit a crime? If so, then our pilot's in trouble.

The ideas of "relevant" similarities and "more" similarities are pretty much non-starters. You need to figure out what the principle is that links or separates

---

21. Edward H. Levi, *An Introduction to Legal Reasoning* (Chicago 1949).

the various cases. Ronald Dworkin,[22] maybe the subsequent generation's Levi, gave some help on this, a kind of clue. He says what you do when you're engaging in legal reasoning is you put the previous decision in the best constructive light. You try to make the best sense out of it. So Dworkin says analogy without theory is blind. An analogy is a way of stating a conclusion, not reaching one, and theory must do the real work, where theory is the principle that links cases or that separates them. The upshot of this is that in any case that's a real case, to figure out whether something's analogous to something else, you have to generate a principle by which the two cases get linked or separated. Lists of factors will be a start, better than Westlaw and LEXIS, but they won't be analogical reasoning. That's not what analogical reasoning is.

To make progress here, we shouldn't give up on artificial intelligence and its potential. A lot more can be done. Good reasoners are going to deal with our copilot case. Can our copilot be fired? Probably anyone in this room, given ten minutes, could figure out ways of thinking the copilot should win or ways of thinking the copilot should lose by reference to the previous cases, reporting on the bank's violations on one hand and the person refusing to commit a crime on the other hand. But how can we make better progress? One thing we might consider is empirical: What are the consequences if you give copilots a right not to fly planes that they see as dangerous? If they can't lose their jobs for that, you might ask, is that going to make people safer? If so, that's a point for the copilot. If copilots do get this right, if the right is given to them, is this going to make it very much harder to run airplanes? Is this going to decrease convenience and order for airplanes and passengers? Those are empirical questions, which Judge Posner, in the relevant case,[23] thought relevant. He's surely right on that. To do the analogical job well, one thing to pursue is these empirical questions— not empirical in the sense that Professor Ashley suggested, not about collecting cases and factors, but an empirical inquiry into the real world effects of one or another legal rule. There's no reason in principle that a computer can't be helpful with that.

If we're not going to get empirical, then what we'd want to do is square our judgment of principle about whether the copilot should win with the rest of the things we think in imaginable cases. Then we'd have to be very creative and go beyond the cases at hand, the precedents, and hypothesize lots of analogies and think what makes best sense of our system of labor law insofar as it bears on this. A lot of really good judges go that route. So far as I can tell from Professor Ashley's really quite outstanding book,[24] HYPO isn't able to do that. What HYPO can do is come up with cases, and it can be pretty exhaustive in that, telling how they might be similar and how they might be different, but in a kind

22. Ronald Dworkin, *Law's Empire* (Belknap 1986).

23. *Buethe v Britt Airlines, Inc,* 787 F2d 1194 (7th Cir 1986).

24. Ashley, *Modeling Legal Argument* (cited in note 2).

of blind fashion, one that is not alert to the need for a guiding principle that might justify a claim of similarity. HYPO can't do what needs to be done.

The upshot of all of this is that artificial intelligence in the current state of the art can be a wonderful advance over LEXIS and Westlaw. What Professor Branting suggested seems to me quite convincing—that this can be a real aid. It's not so much different in the analogical domain from a computer program that can just tell you what the rule or law is. That's very good. What can't be done yet is to do analogical reasoning—to do what lawyers, at least decent lawyers or judges, actually do.

Two qualifications with which I'll end. It may be that in some domains, and I bet that trade secrets is one, you can generate cases that are so sharply hemmed in by precedents that if you look at the precedents in even a kind of crude way, without any principles, you're going to know all you need to know. In some trade secrets cases the fact pattern in question will be one which is not plausibly distinguishable from the precedents. In a case like that, HYPO, or your computer program, is going to do all of the work. In a case like that, by the way, Westlaw and LEXIS are going to do all the work. It's just going to take a little more time with Westlaw and LEXIS than it would with HYPO.

The second qualification seems to me more interesting for the future. There's no reason, so far as I know, in principle to think that in the long run computers won't be able to make the empirical and principled judgments that a good analogizer has to make. The co-panelists would know a lot more about that. To ask whether the social consequences of one or another rule would be A or B, why can't a computer do that? No reason not. So too for generating good normative principles. If a computer can win chess games against pretty good chess players, why couldn't they do that too? If they're doing that, then they're engaging in legal reasoning. Not yet. [*applause*]

MARGOLIS: I'm allowed to ask a question and it happens to have a certain kinship to what Cass was talking about, so I'll ask it and then we'll ask the two other speakers to comment. I have occasion to caution my students against what I call logical democracy. And logical democracy is you just list the arguments on one side and you list the arguments on the other and you count them up and majority wins. And the reason why that's so pernicious is sometimes some arguments are really good and there's a large number of bad arguments on the other side. And so I share Cass's uneasiness at when the AI systems will be ready for that kind of judgment. Why don't you comment?

ASHLEY: Well, I'd like to first respond to something that Cass said and that is that although HYPO's arguments might look like lists of factors, in my work and in the work of other people in AI law, we have been moving in a direction that Cass, I think, would approve. For instance, in my example with the argument that CATO generated, it wasn't just factors but it was reasons why the factors mattered to the legal claim. So I am connecting factors to reasons, and

the arguments are working with those reasons. In work that's being done by colleagues in Europe, Henry Prakken and Giovanni Sartor, they are representing values, principles that are at stake in cases, and those are being worked into the arguments as well. So at least these concepts, these normative concepts, are being worked into the arguments.

Now whether judgment is being applied is another question. But I will opt in favor of the weak AI approach. My game is, I think, to try to find ways in which representing knowledge drawn from our models of how we reason in law, how can that knowledge be applied to do a better job of doing those weak tasks, like retrieval of the right cases at the right time or, for the cases that are retrieved, highlighting what's interesting about them, what's useful about them in the context of an argument. My game is not to try to reproduce the hard tasks of legal reasoning so much as to try to use the knowledge of how we do the hard tasks of legal reasoning to try to build better tools to support those tasks.

We, I think, have been careful not to compare strengths of arguments in terms of numbers. In HYPO there was no comparison of numbers. It was comparisons of sets in terms of set overlap, which is quite a different thing. And we're also, many of us, very sensitive to any attempts to assign numerical weights to anything like principles or values or factors or whatever and to collapse pluses and minuses in that way. We tend to eschew that kind of thing. So I think that we are sensitive to these concerns. We're just gradually working our way upward into the more complicated kinds of arguments that Cass and Howard are talking about.

BRANTING: I guess I'd have several responses. First of all, the legal expert systems that I was describing would clearly all fall into the weak category. They are useful tools. They may be performing some functions other than just weighing arguments. For example, document drafting and prediction are somewhat different tasks. But I guess I would want to emphasize also that people who work in the field of artificial intelligence and law by and large are sensitive to the fact that analogical legal reasoning is not a mechanical process and that the current computational models don't do an adequate job including the evaluative factors that Professor Sunstein pointed to.

On the other hand, in my view, what artificial intelligence is about is really two things. One is making useful artifacts. But a second thing is self-knowledge. That is to say, for example, not that this is what you said, but to say that certain kinds of problem solving or certain kinds of analysis cannot be modeled is kind of a way of saying we can't know ourselves, we can't understand how we solve certain problems well enough to define it with a specificity that's required of a computer. So from my point of view, one of the benefits of this field is that the exercise of trying to formalize legal knowledge in a computer-executable fashion is it forces you to make explicit every piece of knowledge that goes into that decisionmaking and can sometimes make obvious the gaps. For example, some

of the more naïve views of legal problem solving are that it is very rule-driven. One of the ways to demonstrate the inadequacy of this naïve view is to code up some rules and observe that the resulting system doesn't reason anything like a lawyer. I guess my last comment is that I think that the AI and law field progresses through criticisms of the sort that Professor Sunstein just made. The process we'd like to see is: "Here's the argument that's generated by my system and what's wrong with it?"

MARGOLIS: You wanted to add to that?

ASHLEY: I had one thing to add to what Karl just said, and that is that once you have taken the trouble to build a computational model of some interesting phenomenon of legal reasoning, you have a program that works on a range of examples and you can use that as a framework for investigating what that program can't do. So if Cass comes forward with an example of a kind of reasoning that it cannot do, I'm in a position to start playing with the model, to tweak it, change it, see how it has to be revamped or modified. Thus, AI is a kind of empirical methodology for making progress on modeling the phenomena that we all think are so interesting and important.

MARGOLIS: Let me offer Cass just a moment for a comment on a comment on the comments if he wishes, and then open it to the floor.

SUNSTEIN: I want to hear what the audience has to say. There's a computer in the back also that I know has a question. *[laughter]*

MARGOLIS: Do we have any questions? Yes, please.

AUDIENCE: I have a question or a suggestion, and I wanted to get your response, primarily from Mr. Branting and Mr. Ashley. What would you say to the charge that what you are proposing would be the worst thing for legal reasoning, with LEXIS and Westlaw being the second worst, because whatever you produce will ratify the weaknesses of the person doing the inputs. Here is what I mean. With LEXIS and Westlaw, one of the problems, I think, that's happened is that while you can retrieve large numbers of cases based on the contexts or words that you enter, what happens is that you only produce those cases that happen to match the concepts or terms that you were smart enough to pick. And so you get enough case law to produce the set of precedents or a legal reasoning argument that will end up being pretty good. The only dilemma being that you get nothing else that might have made some kind of analogical reasoning that might have given you additional terms you should have looked for.

It seems like the same kind of thing could happen here, because whether you are using factors or another means of deconstructing a case before inputting into the computer, someone has to make a judgment about what factors are worth mentioning. And somebody conceivably could input ten personal injury cases and put in all the facts, and they could all be the same slip and fall case and never think that it was worth mentioning whether the plaintiff was black or white. What would happen if, in the end, the case was that all of the black plain-

tiffs lost and the white plaintiffs won? That model would never show anything like that. So that is just one example that what somebody's judgment is during the inputs—whether it is the person searching or applying the law or the person putting in the tools and items in the computer or database that makes it all work—is going to limit whatever is produced. We might be better off going back to paper and *Shepardizing*, because you have to look at a certain case and come to a judgment of whether or not it is worth seeing.

ASHLEY: I think that that is a very interesting observation. I have to worry about that a lot because I am building, among other things, tutoring systems to teach law students to make arguments with cases. And the fact is that if you model something and focus their attention on one thing, you cause them not to focus their attention on something else. In addition, it is very difficult, because no matter how smart my model is, the students are infinitely smarter. It is very difficult for my program to provide feedback because it is always possible that the student has caught something that the computer simply doesn't know about because it hasn't been modeled. And thus if it says, "No, that's wrong," that's a problem.

It is also a problem even with the graphics that one might use to represent an argument. I didn't show you an example of a claim lattice, but there are ways of graphically representing arguments. We find them useful. They focus your attention on one interesting aspect of an argument but, again, one loses touch with others. It is a design problem that I am constantly trying to finesse.

But as far as the missing of the certain factors that the law does not treat as relevant but really are relevant, that is something that I haven't addressed. I like to look at the work of people like Professor Ted Eisenberg[25] at Cornell, who has done these interesting statistical models of large areas of case law. His students are doing something like my students are doing when they prepare summaries of cases. They go to the cases, they have a form that they fill out for all of the stereotypical facts, but they include the legally "irrelevant" ones as well as the legally relevant ones, to the extent that one can determine them. And then he does statistical analysis on it and comes up with very interesting things. It is a kind of work and a kind of analysis that has to be done as well as the sort of stuff that I am doing. So I agree with you and I try to work around it to the extent that I can.

MARGOLIS: Another question or comments from the other panelists?

BRANTING: I'd be happy to respond to that as well. The perspective that I have been taking is legal expert systems as a vehicle for the delivery of legal expertise. And the question which you have raised is not so much about the adequacy of the vehicle itself, but its content. And of course that is true, it is only going to have as much value as the value of the expertise that is put into it. As

---

25. See, for example, Theodore Eisenberg and James A. Henderson, Jr., *Inside the Quiet Revolution in Products Liability*, 39 UCLA L Rev 731 (1992).

far as the quality of advice or analysis that a system is able to produce, the question about describing the facts to a computer system, I think that that is a separate issue, one that I did not have time to get to.

I listed a number of the tasks that go on in even the most garden variety interaction between an attorney and a client. One of them, the very first one, was problem formulation, and when I was listing the things for which we have good computer models, I didn't include that one. I think we don't have a good computer model of this process of taking a sort of undirected narrative by someone who isn't familiar with legal concepts and reformulating it into a fashion that can be manipulated by one of these legal models. What that means is that the consumers of legal expert systems are initially going to fall into two categories, it seems to me. One category is comprised of lawyers who are able to perform this problem formulation themselves. And the second includes people whose problems are extremely stereotyped and for whom the problem formulation is extremely simple. For example, people who want to know advice about, let's say, social security benefits or, maybe, domestic relations. How do I get a divorce? Can I get a protection order?

AUDIENCE: Computers are progressing very quickly of course, and I was just wondering if you could say very briefly, not what the goal is in the next five years, but what the goal is in the next thirty years, the next fifty years, and the goals for computers in the future?

BRANTING: The long range goal is for computers to become more like the ones in the movies. That's the goal.

AUDIENCE: I'd like to take the question that arose before and point it in the other direction. I agree that there's always a problem of choosing input, and therefore we will have limited output. Now, if we go back to the strong AI argument, which is let's let these computers take over at least part of what lawyers do, then what Professor Sunstein said, if I understood correctly, is that there are some things which simply cannot be done with computers. But, are there things which most lawyers, and I'm not talking about the best lawyers, I'm not talking about professors who have the best knowledge and some quantum leaps of thinking, but most lawyers . . . Most lawyers have a very regular way of approaching things. Most lawyers deal with a limited practice of law. They deal with limited case law of which they are aware of or are willing to search for. Therefore these programs, maybe not today, but in the near future, will probably be able to outperform at least the low end of what lawyers do today. If you take that with the economic arguments of if I'm going to buy from the low end of the food chain, then I'm going to get some of these results, wouldn't it be better to pay less, get a computer program which will probably not do the best but will do at least as well as the lawyer I would probably go to anyway and save money?

SUNSTEIN: Clearly. The issue, I think, is even more interesting than we've gotten a hold of so far. Some of what you say, and some of what's been said,

raises interesting issues about what artificial intelligence can ultimately do and also about what legal reasoning really is. You know, there's a joke among the faculty that you could imagine a computer program—I bet someone could do it—that could write a law and economics article about any topic. Choose a topic and promptly it's the case that you take your favorite or least favorite methodology, and you could imagine a computer program that could do that. I think you're absolutely right that these programs can perform as well as or better than really busy people who don't have time to think a whole lot about what's the best principle to construct for an area of law.

One of the great parts of Professor Ashley's book talks about the relationship between HYPO's performance—HYPO is the computer program—and judicial performance. They're pretty close. That tells us a lot, actually, about the legal system. It shows us that in daily legal reasoning often what does happen is seizing on one or two relevant differences that have been established, not terribly reflectively, as being super salient. That tells us a lot about what our judges are doing. Posner's own opinion in the copilot case[26] was pretty brief. My hunch is that a computer program could improve on it a lot, along one dimension certainly, and maybe in a couple of others. The dimension it could certainly improve on is that it would have access to and use a much larger universe of precedents. Judge Posner just used a couple. The computer could give him a whole lot more. And maybe it could refine the principle by forcing him to grapple with the analogies.

There are intuitive leaps that are involved in chess and in driving. My research assistant, who may be in the room here, found what some of you may not know, that a computer program can drive extremely long distances across the United States at sixty-three miles an hour on average while being able to navigate all but a very small percentage of miles. Now that small percentage is really important if you want to be safe, but it's a small percentage, and what ordinary people would call intuitive leaps for driving, computers can do that.

A lot of creativity in the law, even by people who are very busy, consists of giving a meaning to a case or series of cases that nobody has seen before. This is much more mundane than it sounds, but really thrilling moments in a lawyer's life are when you can create a new pattern out of preexisting materials. That's where creativity lies and in the last few years alone long established cases have been given exceedingly new meanings. Judge Posner, not so recently, but not ages ago, understood the common law as about promoting economic efficiency.[27] No one thought that way before, and it gave a whole new meaning to a tremendous pattern of cases. For artificial intelligence really to take off in law, and this probably isn't short term, it would have to become capable of being a little like a literary critic reading a poem, not in the sense of making nonsense

---

26. *Buethe v Britt Airlines, Inc*, 787 F2d 1194 (7th Cir 1986).

27. See Richard Posner, *A Theory of Negligence*, 1 J Legal Stud 28 (1972).

out of it but in creating a new pattern to it that you didn't see before. That's what even daily lawyers, not the people that are just trying to find out what the law is, but people who actually litigate, that's what they do.

MARGOLIS: We have time for one last question.

AUDIENCE: You were talking about models of analogical reasoning. I was wondering if anything has been done with, say, models of statutory interpretation or constitutional interpretation?

ASHLEY: There has been a lot of work and a lot of progress in representing bodies of statutory rules in a computable way. There has not been very much progress in what we would call statutory interpretation. So, for instance, one sees in civil law jurisdictions people drawing inferences from the structure of the code, for instance, about the meaning of a statutory predicate. We haven't come close to that, I don't think, in AI and law. Also, we have not succeeded in representing the alternative policies that the legislature must have had in mind for a particular statutory provision and trying to look at a problem situation through the statute in light of these alternative policies, actually modeling that. We'd like to, but I don't believe we have yet.