



A Case-Based Approach to Modeling Legal Expertise

Kevin D. Ashley and Edwina L. Rissland
University of Massachusetts

Law is an excellent domain for studying case-based reasoning. Expert system designers use case-based reasoning to capture expertise in domains where rules are ill-defined, incomplete, or inconsistent. As an indispensable supplement to reasoning deductively with legal rules, attorneys and judges reason analogically with precedent cases; rule predicates are simply not sufficiently well-defined for them to infer correct decisions deductively. In fact, one "right answer" seldom exists to legal questions. Legal experts make competing arguments instead, pitting conflicting interpretations of cases and facts against each other.

Hypo, a computer program that performs case-based reasoning in the legal domain, helps attorneys analyze and make arguments about new fact situations in terms of the most relevant precedent cases. To perform this task, indexing and retrieving relevant cases are not enough. Hypo must make factual comparisons of cases relative to the problem situation and determine the legal significance of comparisons in terms of arguments about the problem situation.

We will describe techniques that Hypo employs to

compare cases, choose the best cases for evaluating, and construct arguments about a new fact situation. In particular, we will demonstrate how Hypo critically compares a problem situation (called the current fact situation, or cfs) to the most relevantly similar precedent cases (called most-on-point cases, or mopcs) to outline an argument regarding how to decide the cfs based on its significant similarities to and differences from mopcs.

Hypo's main tool for this task is the "claim lattice" mechanism. A claim lattice projects the case knowledge base (CKB) onto the problem situation to create a neighborhood of cases surrounding the problem situation in which the above comparisons become explicit. We will present a detailed example of a claim lattice actually generated by Hypo to analyze a real legal case.

Neighborhoods of cases

Hypo has a CKB of leading cases in trade secrets law and a dimension-based indexing scheme for retrieving cases that share important features with a problem situation.

Beyond indexing and retrieving relevant cases, however, Hypo must also organize cases in terms of their relevance for making or responding to arguments about problem situations. Attorneys cannot adequately evaluate problem situations without drawing certain symbolic connections among relevant cases in terms of their comparative

(1) Factual similarities and differences relative to the cfs: Which cases are more on point (that is, share more features with the problem situation) and which cases are less on point than other cases?

(2) Outcomes: Of the cases that are most on point to the problem situation, how do their outcomes compare? If comparably on-point cases have conflicting outcomes, what are their differences relative to the problem situation?

(3) Uses: In a legal argument about the cfs, which cases make stronger legal points for a party on a claim? Which are troublesome to an argument? Which distinguish other cases? And which take a point to extremes?

(4) Potential relevance to the cfs: Which cases are nearly on point or most on point but for some missing facts that might jog the attorney's mind to look for and discover new facts and arguments about his cfs?

(5) Significance to other parts of the legal argument: What are the cases and connections when the cfs is viewed from the slant of a different kind of legal claim?

(6) Possible variations: What new cases and connections come into view when the facts of the cfs are changed hypothetically? How do the connections change when features are added, subtracted, exaggerated, or combined with those of neighboring cases?

In effect, an attorney conceptually organizes relevant cases into a neighborhood of cases more or less on point to the problem situation, some with desirable outcomes and some not, some especially useful for responding to others and some only potentially so. Within the context of the relevant-case neighborhood, small changes in problem situation facts can push the situation closer to one precedent and farther from others, with significant results in the ways that cases can be cited and responded to in arguments.

Hypo uses claim lattices to represent the conceptual neighborhood of cases. Claim lattices allow Hypo to exploit connections among relevant cases arising from comparisons relative to the cfs. A claim lattice projects the CKB onto the cfs to create the neighborhood of cases surrounding the problem situation. The case neighborhood is the CKB as seen modulo the cfs. To analyze the cfs, claim lattices (1) organize and abstract all relevant cases from the CKB, (2) place them in a graph according to their comparative similarities and usefulness as precedents for arguing how to decide the cfs, (3) focus on troublesome contrary cases, and (4) suggest fruitful combinations of facts for new hypotheticals.

Knowledge representation in Hypo

Hypo uses two kinds of domain knowledge to construct claim lattices: (1) the CKB containing actual legal cases, and (2) the library of dimensions. Hypo's current CKB contains 30 or so cases on trade secrets misappropriation and a few related areas including basic contract law. Our references provide a complete description of the Hypo program.¹ Other researchers place less or no emphasis on representing and indexing cases and hypotheticals, instead employing generalized fact patterns,² if-then rules,^{3,4} or formal logic.⁵ McCarty's early work emphasized adversarial reasoning and hypothetical reasoning.^{6,7} More recently, he has focused on deontic logic for representing permissions and obligations.⁸

Each legal case in the CKB corresponds to a real legal dispute, tried by a court, whose decision as reported in a published opinion is represented in a case representation language consisting of hierarchical clusters of frames (implemented as flavors) describing main components of the case including plaintiff (*pi*), defendant (*delta*), legal claim, prevailing party (plaintiff or defendant), holding, and facts. Some factual features are in turn expanded and represented as frames (for example, plaintiff, products, employees, disclosure events, secret information, and agreements); our references provide examples.^{1,9} A legal claim is a recognized kind of complaint for which courts will grant relief (for example, breach of contract, negligence, trade secrets misappropriation, and copyright infringement). A holding is the court's decision as to the legal effect on each claim of the case's facts, either in favor of the plaintiff or defendant.

A cfs is represented in the same case representation language as the cases in the CKB. From this basic level of a fact situation's representation, Hypo computes whether certain higher level descriptors called factual predicates are satisfied. Factual predicates state whether or not a particular legal fact is true (for example, "there exist disclosees" and "employee has switched employers") in a fact situation. Factual predicates form a language used to encode the second source of legal knowledge in Hypo; namely, the dimensions.

Dimensions capture the legal relevance of a fact cluster to a claim's merits. For a particular kind of case, dimensions generalize collections of facts that constitute strengths and weaknesses in a party's position. Each of the generalizations can be backed up by one or more cases where a court held in favor of a party, in part because of the fact cluster associated with the dimension. Dimensions allow Hypo to view the cases from various perspectives. They can be thought of as "cross sections" of a case's facts, emphasizing important features from the viewpoint of a

The plaintiff's position is strengthened to the extent that

Brought-Tools: The plaintiff's former employees brought the plaintiff's notes, diagrams, tools to the defendant.
Competitive-Advantage: The defendant's access to the plaintiff's secret information gave the defendant a competitive advantage.
Disclose-Secrets: The plaintiff did not voluntarily disclose his secrets to outsiders.
Noncompete-Agreement: The plaintiff's employees had entered into nondisclosure agreements.
Bribe-Employee: The defendant bribed the plaintiff's employees to switch employment.
Vertical-Knowledge: The plaintiff's secrets were not simply about customer business methods.

Figure 1. Sample dimensions and related factual strengths.

An attorney's corporate client (IBM) complains that Telex has been misappropriating trade secrets with respect to IBM's Merlin disk-drive system. Specifically, IBM complains that

- Telex offered IBM's Merlin project engineers large salaries, stock options, and bonuses (one for \$500,000) as inducements to join Telex (Bribe-Employee).
- All former IBM employees had entered into nondisclosure agreements with IBM to keep IBM's trade secret information confidential (Noncompete-Agreement).
- Because of its access to IBM trade secrets, Telex developed its competing products in substantially less time and at lower expense (Competitive-Advantage).

Figure 2. The cfs based on Telex vs. IBM.

Applicable factual predicates: exists-corporate-claimant, exists-confidential-info, employee-switched-employers, and so forth
Applicable dimensions: Agreed-Not-To-Disclose, Bribe-Employee, Competitive-Advantage
Near-miss dimensions: Brought-Tools, Disclose-Secrets, Vertical-Knowledge
Potential claims: Trade secrets misappropriation, breach of nondisclosure agreement
Relevant CKB citations: Midland Ross, Data General, Structural Dynamics, Raycorp vs. Tronic, Modern Controls

Figure 3. A case analysis record for the cfs.

particular legal claim.

At this point, we have implemented 13 dimensions in Hypo, although we know about 30 dimensions in all for the trade secrets and related-contracts domains. We do not compile these ourselves, but take them from scholarly analyses and treatises.^{10,11} Figure 1 shows examples of some potential strengths and weaknesses in a trade secrets situation and dimensions that capture them.

A dimension is also a frame-like knowledge source. It has several facets enabling Hypo to perform various tasks: First, the program can test if a dimension applies to a case or is a near miss using the dimension's prerequisites (stated in terms of factual predicates). For instance, the prerequisites of the Bribe-Employee dimension are that two corporations (plaintiff and defendant) compete with respect to a product; the plaintiff has confidential product information to which the defendant has gained access by luring the plaintiff's former employees to work for the defendant and

to disclose that information. Second, Hypo compares cases along a dimension using the dimension's focal slots. The Bribe-Employee focal slot is what the defendant offered the plaintiff's employees to lure them to switch employment; its range is a set of possible enticements including salary increases, stock options, bonuses, promotions, or no enticement. To strengthen the plaintiff's position in a fact situation to which this dimension applies, add more enticements for inducing employee defection. Third, Hypo finds similar cases by retrieving cases that the dimension indexes. Bribe-Employee indexes at least two cases in the CKB: (1) Telex vs. IBM (in which the court held for plaintiff IBM's trade secrets claim where Telex had offered stock options, higher salaries, and bonuses—one for \$500,000—to IBM's employees); and (2) the Midland Ross case (in which the defendant won, even though the former employee gained a modest salary increase in switching employers). Our references provide other examples of dimensions.^{1,12-14}

ated by the Hypo program for analyzing Figure 2's cfs from the viewpoint of a trade-secrets-misappropriation claim (a separate claim lattice exists for each possible claim.)

Primarily, claim lattices order relevant cases from the CKB in terms of how on point they are to the cfs, which is measured by the degree of overlap between cfs dimensions and those of retrieved cases. In other words, the claim lattice captures the extent that cases share cfs strengths and weaknesses. More specifically, of all dimensions in the cfs D-list, only those in a particular successor node's D-list (a subset of the cfs D-list) apply to each case associated with that node. Each successor node is the ancestor of all nodes whose dimensions are proper subsets of the successor node's subset. Claim lattices order these descendants according to the inclusiveness of their subsets of the cfs D-list. Each successor node may (1) contain pointers to more than one case, and (2) comprise cases won by plaintiffs and defendants.

Interpreting the claim lattice

A claim lattice's ordering scheme enables the lattice to capture a sense of closeness to the cfs of cases in the CKB, viewing relevant cases in terms of their "on-pointness" to the cfs. Computationally, this means that nodes closest to the root can be considered mopcs to the cfs (as long as their subsets of the cfs D-list do not contain near-miss dimensions); leaf nodes are the least on point. All displayed cases are relevant to the cfs because they all share some legally important strengths or weaknesses with the fact situation as represented by dimensions shared with the cfs.

Hypo judges dynamically which cases appear most on point. Even well-known landmark cases will not appear if they are not relevant to the cfs; for instance, if they don't share enough overlap of dimensions (and therefore facts), their decisions rest on factors not present in the current case. Likewise, a seemingly unimportant case might be very relevant modulo the cfs because it not only shares the same base of facts (and therefore dimensions) but it might also share more than any other case. This is shown in the claim lattice by positioning nearer to the cfs those cases sharing more dimensions.

Different major branches of the lattice correspond to different ways to argue the case, one way for each branch. Let's hypothesize that the user attorney seeks an argument for the defendant in Figure 2's cfs. Hypo can argue the case for the defendant by citing a pro-defendant mopc. In the cfs claim lattice (Figure 4), the mopcs are Raycorp vs. Tronic, Modern Controls, and Telex vs. IBM. (Of course, Telex vs. IBM is also the basis of Figure 2's cfs. A case should be most on point to itself!) Of the three mopcs, only

Raycorp vs. Tronic held for the defendant (*delta*), so Hypo cites that case in its argument. Midland Ross is not an mopc because, although it is very close to the root and was won by the defendant, the Disclose-Secrets dimension—which applies to Midland Ross and which would help the defendant if it applied to the cfs—is only a near miss for the cfs (as indicated by the asterisk next to Disclose-Secrets in Figure 4).

Hypo produces the following argument, which favors the defendant as Side 1 on a claim for trade secrets misappropriation in the cfs:

[A] Side 1's point (defendant)—Cite Raycorp vs. Tronic: The defendant should win even though the employee who disclosed information signed a nondisclosure agreement with the plaintiff.

[B] Side 2's response (plaintiff)—Distinguish Raycorp vs. Tronic: In the cfs, the defendant's access to the plaintiff's product information saved it more time or expense than in Raycorp vs. Tronic and the defendant paid the plaintiff's former employee to switch employment.

Cite Modern Controls as counter example: This case is just as on point and held for the plaintiff where the employee who disclosed information to the defendant signed a nondisclosure agreement with the plaintiff.

In support of its point, Hypo draws the analogy between the mopc and the cfs. Since mopcs share the most legally important strengths and weaknesses with the cfs (that is, mopcs are the closest analogies to the cfs), Raycorp vs. Tronic is the most persuasive case Hypo could cite for the defendant as Side 1. Relevant similar facts are just those summarized by the dimension(s) that the mopc shares with the cfs. This case has but one shared dimension (Agreed-Not-To-Disclose). The argument point in [A] is not very strong, but it is well formed from a legal viewpoint.

Hypo also uses the claim lattice to figure out how to respond to an argument like Side 1's by citing counter examples to the cited case. In a legal argument, an attorney can discount the persuasive effect of a cited case by citing another case that is equal to or more on point but that held for the opponent. Finding such a case is a simple matter in the claim lattice; if it exists, it must be an mopc residing in one of the nodes between the root and the node containing the cited case (mopcs on other major branches of the lattice won't do—that would be like comparing apples and oranges). Mopcs in Figure 4 occupying positions as counter examples are Telex vs. IBM, and Modern Controls, both of which held for the plaintiff (*pi*) and are equal to or more on point than Raycorp. In other words, Hypo makes Side 2's (the plaintiff's) response to Side 1's argument by citing the Modern Controls case as a counter example (see [B] above). It would be silly for Hypo to cite the Telex case in response because, for illustration, we have chosen Telex as

***Claim lattices
suggest interesting
hypothetical cases.***

the cfs. In supporting the response, Hypo underscores additional facts (if any) that make the counter example equal to or more on point than the case cited for Side 1.

The claim lattice supports other ways of responding to a point by distinguishing the cited case. Suppose Hypo cited the Midland Ross case on behalf of Side 1, as in the following exchange:

[C] Side 1's point (defendant)—Cite Midland Ross: The defendant should win even though the defendant paid the plaintiff's former employee to switch employment.

[D] Side 2's Response (plaintiff)—Distinguish Midland Ross: In the cfs, the defendant's access to the plaintiff's product information saved it more time or expense than in Midland Ross, and the plaintiff's former employee entered into a nondisclosure agreement with the plaintiff. In Midland Ross, the plaintiff disclosed its product information to more outsiders than in the cfs.

In distinguishing, Hypo points out the "dis-analogy" between the cfs and Midland Ross by reciting unshared dimensions that help the plaintiff in the cfs or hurt the plaintiff in Midland Ross. Hypo points out that the facts associated with the Disclose-Secrets dimension obtained only in Midland Ross and not in the cfs (namely, that the plaintiff disclosed its secrets to outsiders). Also, in the cfs, the defendant gained a competitive advantage and the plaintiff's employee had entered into nondisclosure agreements. In other words, using the claim lattice, Hypo knows that (and why) the cfs presents a much stronger case for the plaintiff than does Midland Ross.

As the above examples illustrate, the claim lattice embodies (1) Hypo's knowledge of how to compare cases relative to the cfs, and (2) what the legal significance of those comparisons are in terms of arguments about the cfs.¹²

Hypo uses that knowledge to help attorneys build arguments; for example, by finding cases that, potentially, are most on point. A potential mopc closely resembles the fact situation, except that some dimensions (that is, strengths or weaknesses) that apply to it are near misses with respect to the cfs; they are located in nodes closest to the root whose subsets of dimensions contain near misses. Potential mopcs are useful (1) when fact-finding about the cfs, (2) when looking for cases to cite as counter examples to a particular mopc, and (3) when planning for the contingency that an opponent might be able to prove that missing facts are true. Figure 4's Midland Ross, Structural Dynamics, and Automated Systems cases are all potential mopcs. If it were true that the plaintiff in the cfs had disclosed his secrets to (let's say) 200 outsiders, Midland Ross would become a most important case to Side 1, allowing the defendant's point in [C] to be recast as follows:

[C'] Side 1's point (defendant)—Cite Midland Ross: The defendant should win where the plaintiff disclosed secrets to outsiders even though the defendant paid the plaintiff's former employee to switch employment.

[D'] Side 2's response (plaintiff)—Distinguish Midland Ross: In the cfs, the defendant's access to the plaintiff's product information saved it more time or expense than in Midland Ross and the plaintiff's former employee entered into a nondisclosure agreement with the plaintiff.

The point in [C'] is stronger than in [C] because the analogy between the cited case and the cfs is stronger. Correspondingly, the distinction in response [D'] is weaker than that in [D].

Alternatively, if the cfs were more like the potential mopc, Structural Dynamics, Side 2 would have an even stronger response to the point in [A], where it can cite the following more-on-point counterexample:

[B'] Side 2's response (plaintiff)—Distinguish Raycorp vs. Tronic: In the cfs, the defendant's access to the plaintiff's product information saved it more time or expense than in Raycorp vs. Tronic and the defendant paid the plaintiff's former employee to switch employment.

Cite Structural Dynamics as counter example: This case is more on point and held for the plaintiff where the employee who disclosed information to the defendant signed a nondisclosure agreement with the plaintiff and brought the plaintiff's product development tools to the defendant.

The claim lattice also focuses on another kind of extreme case; namely, boundary cases in which the plaintiff still won (or lost) despite the fact that, of all cases in the CKB, they presented the weakest (or strongest) fact situations for the plaintiff along some dimension. A boundary case may not be as on point as a mopc, but it is still useful to Hypo as a counter. For instance, suppose in the cfs that IBM had made disclosures to 200 outsiders and that Side 1 had then cited the Midland Ross case, which would be a mopc for the defendant. In responding for Side 2, Hypo could also cite the following counter example (see Figure 4):

[D'] (continued) Side 2's response (plaintiff)—Cite Data General as counter example: This case held for the plaintiff even though the plaintiff disclosed to 6000 outsiders (more than in the cfs or Midland Ross).

The new response implies that, even had the plaintiff's behavior been much worse, it still should win. Here, Data General is used as an example of a much weaker case for

the plaintiff along the Disclose-Secrets dimension (with disclosures to 6000 outsiders), but whose outcome still favored the plaintiff.

In the previous example, Hypo posed a hypothetical variation of the cfs (that IBM had made disclosures to 200 outsiders) to make a legal point about the connections between the Data General and Midland Ross cases. That is a natural thing for attorneys to do, and points out one of the most important uses of claim lattices: They suggest interesting hypothetical cases.

The claim lattice structure provides clues about how to "flesh out" sparse areas of the CKB, and suggests how to modify hypothetically either (1) actual cases, or (2) the cfs for constructing hypothetical cases and new connections between the cfs and its neighbors. Having located the cfs in its neighborhood of cases, Hypo uses hypothetical modifications to explore the neighborhood.

Interesting hypotheticals reside in two locales of the claim lattice: First, they reside in nodes containing potential mopcs, as the above hypothetical variant of the cfs suggests. Second, they reside between nodes containing mopcs.

Hypotheticals residing in nodes. Using potential mopcs as targets, the cfs can be modified slightly to incorporate "missing" facts associated with near-miss dimensions; in Figure 4, for example, (1) by adding disclosures as in Midland Ross, (2) by having former employees bring the plaintiff's product-related notes and copies of code as in the Structural Dynamics case, or (3) by making the secret information be vertical knowledge about the customer's business methods as in the Automated Systems case. Our references provide a more complete account of posing hypotheticals in Hypo.^{1,15}

Hypo uses hypothetical variants of the cfs to explain the potential mopcs' significance and to illustrate the hypothetical fact's effect on the argument. For example, if IBM did make such disclosures, not only would Midland Ross become an mopc, but Data General would also become potentially an even more on-point case that could be used to respond to an argument citing the Midland Ross case. Hypo makes these changes in possible arguments explicit by reconstructing the claim lattice around the modified cfs. Using the claim lattice to plan for contingencies in an argument is most practical. In the real *Telex Corp. vs. IBM Corp.*, 367 F. Supp. 258, 358 (N.D. Okla. 1973), defendant Telex actually attempted to raise the issue of disclosures by IBM. In its opinion, the court cited another case involving disclosures by Midland Ross, but disposed of the defense on other grounds.

Hypotheticals between nodes. Other interesting hypotheticals reside between nodes containing mopcs, and

combine features of plaintiff mopcs with those of defendant mopcs. In pitting pro-plaintiff and pro-defendant features against each other, these hybrid mopcs implicitly raise the issue of which features are more important (that is, how such a case would be decided). In Figure 4, an interesting hypothetical hybrid mopc combines pro-plaintiff features of the Telex case with pro-defendant disclosures of secrets as in the Midland Ross case. It happens to be the same hypothetical as above, but now we are changing the Telex case rather than the cfs.

Judges and attorneys actually use hypotheticals like this one. In his opinion in the Executive Development Center of Boston vs. Hirsch, Civ. Action 50441 (Suffolk Superior Ct. Mass. 1985), p. 28, Judge Young makes his point by constructing a similar hypothetical pitting the plaintiff's disclosures against the defendant's improper means of access. The hybrid hypothetical can be made even more interesting by pushing the number of discloseses to extremes (one disclosure or one million).¹⁵

Hypo can also use the claim lattice to view a fact situation from the perspective of other claim types. Figure 4's lattice depicts the cfs as a trade-secrets-misappropriation case. But, by virtue of the Agreed-Not-To-Disclose dimension, the cfs could also be thought of as involving a claim for breach of nondisclosure agreements entered into by employees. The claim lattice representing that claim is yet another projection of the CKB onto the cfs, this time creating a neighborhood of contracts cases organized along different dimensions like Agreement-Supported-By-Consideration but subject to the same interpretive methods as described above.

We have described a technique for dynamically viewing or reorganizing a CKB around a particular fact situation to provide a sophisticated analysis of that fact situation. Claim lattices project the fact situation through the CKB to create a neighborhood of cases, surrounding the situation, in which factual and interpretive comparisons of the cases relative to the situation become both simple and explicit. We have also described how the Hypo program uses claim lattices to (1) create skeletal legal arguments about fact situations, (2) spot troublesome contrary cases, and (3) suggest fruitful combinations of facts for new hypotheticals.

Acknowledgments

This work was supported (in part) by the Advanced Research Projects Agency of the Department of Defense, monitored by the Office of Naval Research under Contract No. N00014-84-K-0017; by University Research Initiative Award No. N00014-86-K-0764; and by an IBM Graduate Student Fellowship.

References

1. K.D. Ashley, *Modeling Legal Argument: Reasoning with Cases and Hypotheticals*, PhD dissertation (COINS Tech. Report 88-01), Computer and Information Science Dept., University of Massachusetts, Amherst, Mass., 1988.
2. A.vdL. Gardner, *An Artificial Intelligence Approach to Legal Reasoning*, PhD dissertation, Computer Science Dept., Stanford University, Stanford, Calif., 1984.
3. D.A. Schlobohm and D.A. Waterman, "Explanation for an Expert System that Performs Estate Planning," *Proc. First Int'l Conf. Artificial Intelligence and Law*, ACM, New York, N.Y., 1987.
4. D.A. Waterman and M. Peterson, "Rule-Based Models of Legal Expertise," *Proc. First Nat'l Conf. Artificial Intelligence*, AAAI, Menlo Park, Calif., 1980.
5. M.J. Sergot et al., "The British Nationality Act as a Logic Program," *Comm. ACM*, May 1986, pp. 370-386.
6. L.T. McCarty and N.S. Sridharan, "The Representation of an Evolving System of Legal Concepts: Prototypes and Deformations," *Proc. Seventh Int'l Joint Conf. Artificial Intelligence*, Morgan Kaufmann, Los Altos, Calif., 1981.
7. L.T. McCarty and N.S. Sridharan, "A Computational Theory of Legal Argument," Tech. Report LRP-TR-13, Laboratory for Computer Science Research, Rutgers University, New Brunswick, N.J., 1982.
8. L.T. McCarty, "Permissions and Obligations," in *Computing Power and Legal Reasoning*, C. Walter, ed., West Publishing Co., St. Paul, Minn., 1985.
9. E.L. Rissland, E.M. Valcarce, and K.D. Ashley, "Explaining and Arguing with Examples," *Proc. Fourth Nat'l Conf. Artificial Intelligence*, AAAI, Menlo Park, Calif., 1984.
10. M.R. Gilburne and R.L. Johnston, "Trade Secret Protection for Software Generally and in the Mass Market," *Computer Law J.*, Spring 1982.
11. R.M. Milgrim, *Business Organizations, Milgrim on Trade Secrets*, Vol. 12, Matthew Bender, New York, 1985.
12. K.D. Ashley and E.L. Rissland, "Compare and Contrast, A Test of Expertise," *Proc. Sixth Nat'l Conf. Artificial Intelligence*, AAAI, Menlo Park, Calif., 1987.
13. E.L. Rissland and K.D. Ashley, "A Case-Based System for Trade Secrets Law," *Proc. First Int'l Conf. Artificial Intelligence and Law*, ACM, New York, N.Y., 1987.
14. K.D. Ashley and E.L. Rissland, "But, See, Accord: Generating Blue Book Citations in Hypo," *Proc. First Int'l Conf. Artificial Intelligence and Law*, ACM, New York, N.Y., 1987.
15. E.L. Rissland and K.D. Ashley, "Hypotheticals as Heuristic Device," *Proc. Fifth Nat'l Conf. Artificial Intelligence*, AAAI, Menlo Park, Calif., 1986.



Kevin D. Ashley is a lecturer on law at Boston University Law School, where he teaches a course on computers and law. This June, he became an academic visitor at the IBM Thomas J. Watson Research Center, where he is pursuing his research interests in developing computer systems for assisting attorneys in law practice and in contributing to AI's understanding of analogical and case-based reasoning, argumentation, and explanation. He received his BA in philosophy (magna cum laude) from Princeton University in 1973, his JD (cum laude) from Harvard Law School in 1976, his MA in 1985 and his PhD in computer science in 1988 from the University of Massachusetts. From 1976 through 1981, he practiced law as a litigator at White & Case in New York City.

This work received the Philips Laboratories award for best student paper at the Fourth IEEE Conference on Artificial Intelligence Applications. The authors can be reached at the Dept. of Computer and Information Science, University of Massachusetts, Amherst, MA 01003.



Edwina L. Rissland is an associate professor of computer and information science at the University of Massachusetts at Amherst. Since 1985, she has been a lecturer on law at Harvard Law School, where she teaches a course on AI and legal reasoning. During the spring of 1986, she spent a sabbatical semester as visiting professor at Stanford's Law School and Knowledge Systems Lab, and was a Fellow of law and computer science at Harvard Law School in 1982-83. Her current research includes case-based reasoning, AI and legal reasoning, intelligent example selection for learning systems, and intelligent user interfaces. Before joining the University of Massachusetts in 1979, she taught mathematics and performed research at MIT. She received her ScB in applied mathematics (magna cum laude) from Brown University, her MA in mathematics from Brandeis University, and her PhD in mathematics from MIT in 1977. She was a staff scientist at MIT's Lincoln Laboratory from 1974 to 1975.