Int. J. Man–Machine Studies (1991) **34**, 753–796

Reasoning with cases and hypotheticals in HYPO

KEVIN D. ASHLEY

University of Pittsburgh, School of Law and Learning, Research and Development Center, Pittsburgh, Pennsylvania 15260, USA

HYPO is a case-based reasoning system that evaluates problems by comparing and contrasting them with cases from its Case Knowledge Base (CKB). It generates legal arguments citing the past cases as justifications for legal conclusions about who should win in problem disputes involving trade secret law. HYPO's arguments present competing adversarial views of the problem and it poses hypotheticals to alter the balance of the evaluation. HYPO uses Dimensions as a generalization scheme for accessing and evaluating cases. HYPO's reasoning process and various computational definitions are described and illustrated, including its definitions for computing relevant similarities and differences, the most on point and best cases to cite, four kinds of counter-examples, targets for hypotheticals and the aspects of a case that are salient in various argument roles. These definitions enable HYPO to make contextually sensitive assessments of relevance and salience without relying on either a strong domain theory or *a priori* weighting schemes.

1. Introduction

Learning from past experience is a hallmark of intelligence. Decision makers faced with new problems, benefit from the experience of having attempted to solve past problems or prior cases. For example, government officials facing a hostage crisis, remember previous disastrous attempts at making deals with hostage takers and adhere to a rule of policy derived from those cases: never make deals with hostage takers. When diagnosing a patient with unfamiliar symptoms, a physician undertakes a battery of tests, most yielding no useful results, until finally a particular test leads to a definitive diagnosis. The next time the physician encounters such symptoms, he first tries the test that lead to a productive result in the prior case. A would-be home-owner in a new city tries to determine if the asking price of a house for sale is reasonable. She or he consults the city tax office's listings to find the sale prices of comparable houses in the immediate neighborhood in order to fix a price based on these past cases.

Comparing current problems to past cases is a common and important way of making and justifying decisions. Frequently, there are no other solution methods at hand. Even if there are alternatives, comparing with past cases is a way of checking the reasonableness of, or explaining a proposed decision.

Past cases, if the decision maker remembers them, exert an effect on the decision making process. They provide a framework for analysing the problem, causing the decision maker to focus on aspects of the probelm that were important in the past. The focus may be mistaken; the past cases may be misleading. In all of the above examples, the decision maker is not guaranteed to make the best or right decision by referring to past cases. For instance, there may be circumstances in the present case

0020-7373/91/060753 + 44$03.00/0

not found before, which would justify making a deal with hostage takers. The case-based justification would then be wrong. Nevertheless, a past case focuses the decision maker on some aspects of the problem. It is instructive to examine how.

As the three examples suggest, past cases bear on current decision making in three broad ways:

(1) generalization;
(2) constrained search; and
(3) comparative evaluation.

Prior cases sometimes support generalizations that can be applied to decision making. A generalization may be a rule that can be applied deductively to solve or classify new problems, or it may be a causal explanation of the result in the case that can be mapped onto a problem analogically. The generalizations identify the parts of a problem that are important and provide a procedure and warrant for reaching a conclusion. In effect, the prior case is a datum from which to infer or derive the generalizations. Depending upon the domain and kind of generalization, the nature of the generalization's support may vary. It could be primarily statistical given a large sampling of positive and negative case examples. Statistical generalizations are computed dynamically from cases in MBRtalk (Stanfill & Waltz, 1986; Stanfill, 1987). The generalizations may be imposed on a domain. The CABARET program represents legal rules whose authority is imposed on a domain by statute but whose technical terms are ill-defined, requiring cases for help in evaluating whether the terms apply (Rissland & Skalak, 1989, 1991). A generalized explanation could be derived from a single explanation example as in the CASEY program (Koton, 1988*a*, 1988*b*), or, alternatively, from a small collection of positive and negative case examples as in PROTOS (Bareiss, 1989; Bareiss, Porter & Wier, 1987), and GREBE (Branting, 1991). CBR programs that employ structured explanations of past cases shade into the next category.

A prior case may constrain the search for a solution by providing a template for a solution. Instead of solving the problem from scratch, which may involve a relatively weakly focused search through the space of operators and bindings, the reasoner first searches a different space of past solved cases for a case similar to the current problem. That search is easier, at least theoretically, because the past cases are indexed. The reasoner simply follows an index entry to a relevant prior solution and then applies and adapts its solution steps to the current problem. In effect, a past case is a compiled solution. An important subcategory of such search-shortening strategies is keeping track of past problem-solving failures, so as to avoid replicating the doomed solution paths. Research in case-based planning provides some classic examples of this solution adaptation approach, for example Kolodner (1983), Kolodner, Simpson and Sycara-Cyranski (1985), Hammond (1986, 1987, 1989). CASEY's explained cases also serve as compiled solutions to diagnostic problems (Koton, 1988*a*, 1988*b*). The PRODIGY program is an extreme example of this approach in which complete histories of the decision-making process for a past case are stored, improving the adaptability of past cases at the expense of complicating the process of saving, finding and applying them (Carbonell, 1986; Carbonell & Veloso, 1988).

Finally, prior cases may provide a means of comparative evaluation. A current

problem may be evaluated by comparing and contrasting it with prior evaluated cases. The reasoner assigns a value, such as a dollar amount, by extrapolating from similar previous cases of known value. For example, he reasons that a house should be priced a little higher than a previous sale to take into account inflation and an extra bath. In effect, the past cases are data on which to base an argument that the current problem has a particular value, either the same value as in the case or one that may be extrapolated from that value. The evaluations need not be dollar amounts, they may be symbolic classifications as well. My work on HYPO is a good example of symbolic evaluation of problems, by comparing and contrasting them with prior cases (Ashley, 1988, 1989, 1990; Ashley & Rissland, 1987, 1988*a*, 1988*b*). CABARET integrates HYPO's case-based evaluation approach with a rule-based approach (Rissland & Skalak, 1989, 1991).

All case-based reasoning (CBR) systems involve some combination of the three methods: generalization, constrained search and comparative evaluation. For example, almost all case-based reasoners employ case indices. In most reasoners, the indices are conceptual indices; they index the cases by concepts abstracted from the cases. Some amount of generalization is inevitable in the construction and use of such indices. Even a "pure" inductive generalization method requires cases for evaluation. Although theoretically, once a rule has been derived the case or cases on which it is based could be discarded, in practice the cases should be retained. The cases are still useful as examples for explaining the rule, as test cases when the rule needs to be modified, and as justifications for the rule's existence. Where the rule's predicates cannot be well-defined, as in many legal rules, the cases are essential for evaluating whether the rule applies to a problem (e.g. Rissland & Skalak 1991; Branting, 1991).

The three methods of generalization, constrained search and comparative evaluation are appropriate for different domain types and tasks. They make different demands on the availability of a strong domain theory for analysing and solving problems.

(1) Generalization: a pure generalization approach assumes the availability of a strong domain theory. Indeed, the rules or explanations derived from the cases are part of the domain theory. To the extent that the rules or explanations can be applied deductively, without further recourse to cases, the domain theory is strong.

(2) Constrained search: a constrained search approach may be useful where there are only weak methods for solving a problem. But even if there are stronger analytic solution methods available, the solutions may still involve a lot of search which may, in theory, be shortcut by employing past cases as solution templates.

(3) Comparative evaluation: by contrast, the comparative evaluation method does not assume a strong domain theory. Comparative evaluation by cases is useful where there are no strong domain theories; it may be the only way of coming up with a value. For example, there is no strong analytic theory for assigning a value to a house from scratch, but there may be some systematic means for estimating its value based on past sales taking into account differences in the house's features and correcting for appreciation since the date of sale.

The different methods also correspond to different tasks. Problems that are likely to have one uniquely right solution are more susceptible to the constrained search

approach or a pure generalization approach. Problems for which there usually is no one right answer but only alternative reasonable answers, are more susceptible to an evaluation approach.

Whatever the approach, a common problem is defining and assessing relevant similarities and differences. Is a case similar enough to a problem? If it is only a partial match, are the differences relevant? How does one select intelligently among a number of competing more-or-less similar cases? Assessing relevance is made more complex because it depends highly on context. Generalizations about the importance of particular features may be unwarranted in the light of a problem's and case's particular facts or the task at hand. Finally, assuming that relevant similarities and differences can be defined and assessed contextually, what inferences does the CBR system draw from comparing problem to case?

Taking HYPO as a "case" in the history of CBR systems, this article describes and illustrates how HYPO's design addresses these questions in a manner appropriate for its domain and task. The main resource for information about HYPO is Ashley (1990). This article provides a comprehensive summary of that work, a detailed extended example of HYPO in action, and a good basis for comparing and contrasting HYPO and other CBR systems' domains, tasks and designs.

2. HYPO's task and domain

HYPO is a case-based reasoning system that generates legal arguments citing prior cases, or precedents, as justifications for legal conclusions about who should win in problem disputes. HYPO's design represents a hybrid generalization/comparative evaluation approach appropriate for a domain with a weak analytic theory and for tasks that seldom involve just one right answer.

The domain is trade secret law. Trade secret law protects the rights of one who has developed competitively valuable information. As long as the owner maintains the information in confidence, the law protects his rights to the information. If anyone uses or discloses the information either in breach of a confidential relationship with the owner or by unfair or illegal means, the owner may recover on a claim of misappropriation of the owner's trade secret. Trade secret misappropriation is one of many kinds of legal claims like copyright infringement, breach of contract, or professional malpractice, each designed to protect various legal personal and property rights from certain kinds of incursions.

Many issues arise in determining who should win a trade secret misappropriation claim: the plaintiff, usually the owner of the trade secret who sues for misappropriation, or the defendant, the party sued, usually a corporate competitor or a former employee. Suppose the owner has disclosed the information to his employee or to an outsider in negotiations, trying to solicit the outsider's interest? Suppose someone learns of the information, not from the owner but by reinventing it him or herself? Suppose the owner has not taken adequate steps to prevent the information from becoming generally known, or suppose the information becomes generally known despite the owner's best efforts? Does the owner have property rights in these situations and is he or she protected from the defendant's behavior?

Judges and legislative drafters in trade secret law have recognized the difficulty of prescribing rules for deciding all of the issues that arise. A noted authority on trade

secret law has quoted one court as follows: "Another court, less sanguine about the practicalities of defining a trade secret, has concluded that a 'trade secret is not necessarily a secret within the common use of that word. It is at best a nebulous concept which · · · is somewhat incapable of definition'" Milgrim (1985), Section 2·01, p. 2–21. According to the drafters of the Restatement of Torts—an authoritative body of rules and commentary setting out trade secret law and other areas of tort law—"an exact definition of a trade secret is not possible" [Restatement (First) of Torts, Section 757 comment b].

In essence, the drafters of the Restatement provisions on trade secret law recognized that the domain of trade secret law has a weak analytic theory. They took an interesting tack in dealing with this weakness. In order to supplement their definition of a trade secret, they employed factors:

> An exact definition of a trade secret is not possible. Some *factors* to be considered in determining whether given information is one's trade secret are:
> (1) the extent to which the information is known outside of his business;
> (2) the extent to which it is known by employees and others involved in his business;
> (3) the extent of measures taken by him to guard the secrecy of the information;
> (4) the value of the information to him and to his competitors;
> (5) the amount of effort or money expended by him in developing the information;
> (6) the ease or difficulty with which the information could be properly acquired or duplicated by others [Restatement (First) or Torts, Section 757 comment b].

In other words, in order to compensate for their inability to define adequately trade secrets, the drafters of the Restatement set out six factors that attorneys and judges could use to help assess legal fact situations. The approach has caught on. "It is most common to discuss trade secrets in terms of the factors · · · tending to define a trade secret" Nimmer (1985), pp. 3–4.

Factors like the above are generalizations. Unlike rules, they do not specify necessary and sufficient conditions for a conclusion. Instead, they designate collections of facts, commonly observed in cases, that tend to strengthen or weaken a plaintiff's argument in favor of a conclusion, such as a legal conclusion that the plaintiff has a trade secret.

Factors are a kind of expert legal knowledge employed to make up for the lack of a strong domain theory. For each kind of legal claim, authors of legal reference works and scholarly articles identify factors to look for in assessing the strength or weakness of a party's position, as well as cases that exemplify the factors and their effect on a decision. Even though there may be no way to set out the necessary and sufficient conditions of a claim like trade secrets misappropriation, at least knowing factors, one can identify the facts of a situation that tend toward or away from the conclusion that the plaintiff has a winning claim.

In order to solve a problem using factors, the first step is to determine the factors that apply. Typically, a case involves a number of competing factors. Some factors favor the plaintiff's claim; others favor the defendant's. There has to be some way to resolve or combine the effects of the competing factors.

The problem with employing factors to evaluate a dispute is that experts do not agree about the factors' relative significance. Attorneys, for example, are loathe to make categorical statements that a particular factor is always more important. Nor will they often assign a numerical weight or probability that a particular factor will

control an outcome. This reluctance stems, in part, from attorneys' awareness that the significance of a factor depends very much on context. While one factor may frequently be more important than another, there is always a case where the opposite is true because, for example, other factors are present in the case. In addition, attorneys are uncomfortable with numerical or statistical weights. Arguing with statistical summaries of cases is uncommon and suspect where the significance of the numbers is debatable and where there is a strong institutional preference for symbolic arguments.

Without a scheme for numerically combining the effects of factors, evaluating a case is problematic. If factors had numerical weights, judges might simply identify the competing factors and apply some combining function to assess their net effect, for example plus 0·78 or minus 0·36 in favor of plaintiff. Such numbers are meaningless for judges and attorneys as a means of evaluating a problem, not least because they are expected to justify their legal conclusions symbolically by citing some legal authorities in support. Legal authorities include statutes, regulations, court-made rules of law, and precedents. There simply are no legal authorities that support numerical weights assigned to factors.

In practice, precedents provide a mechanism for evaluating legal situations in terms of factors without a weighting scheme. Given a situation presenting a number of conflicting factors, some favoring a plaintiff's claim and some not, one way to justify an outcome is to cite as a legal authority a precedent that involved some or all of the same conflicting factors. The justification says, in effect: (1) in the precedent, a prior court resolved the competing factors in favor of a particular side; (2) the current situation is analogous to the precedent because it involves the same competing factors; (3) therefore, the current dispute should be resolved in the same way.

3. Justifying conclusions with factors and cases

A series of Venn diagrams illustrates how precedent cases can be used to justify conclusions about a problem analysed in terms of factors. The figures illustrate intuitively how precedents justify resolutions of competing factors. HYPO justifies conclusions about legal problem situations, in particular whether the plaintiff should win or lose its legal claim for misappropriation of trade secrets. The factors used in HYPO are similar to those listed above for trade secret misappropriation claims. In principle, however, the approach could be used for any kind of problem classification for which factors can be identified.

For each kind of claim, factors that favor (or disfavor) the plaintiff's side are identified by experts. Figure 1 depicts the legal claim of trade secrets misappropriation and shows all of the important factors (f1) that favor a positive outcome for the plaintiff (+) and those that favor a negative outcome (−). In other words, the negative factors favor an outcome for the defendant. A related claim like breach of a nondisclosure agreement, a specialized kind of contract claim, would involve different, though possibly related, factors.

A problem can be represented as a collection of competing factors, some of which favor a conclusion and some of which do not. Figure 1 shows a problem situation *P* and its set of competing factors. In order to justify assigning an outcome

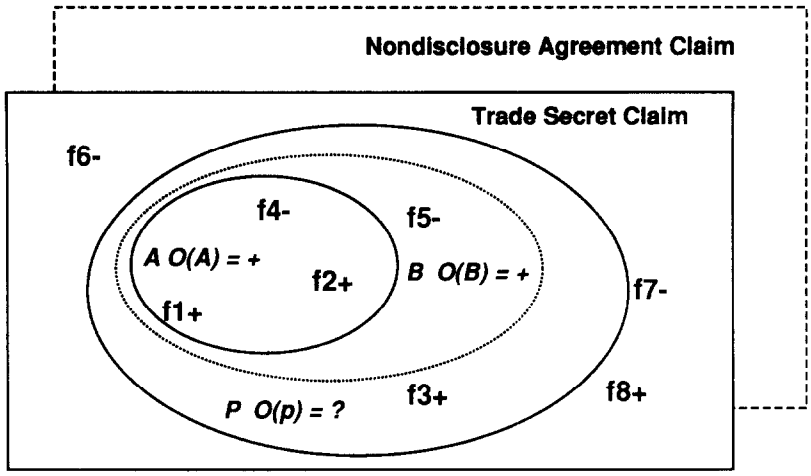

FIGURE 1. Representing claims with factors: each claim, trade secrets and breach of a nondisclosure agreement, has its own factors. A problem situation P is represented as a collection of competing factors. Case A arguably justifies resolving P's factors in favor of plaintiff (+) but can be distinguished on the basis of f5. Case B is more on point than case A. If defendant won case B ($O(B) = -$), B would be a trumping counter-example to case A.

to a problem, $O(p)$, one must justify a resolution of the five competing factors. If there were some numerical weighting scheme or hierarchical ordering of factors, this would be the time to use it. Unfortunately, in law, there are no such schemes.

Instead an attorney seeks to find a past case that involves the same or a similar collection of factors. Precedents may be represented just like problem situations as collections of competing factors. The difference is that a precedent has been assigned an outcome by an authoritative decision maker, the court. Figure 1 shows a case A that involved some of the same factors as problem P where the outcome was in favor of the plaintiff (i.e. $O(A) = +$.) Citing case A, an attorney may argue that problem P should be decided in favor of the plaintiff, just as A was. The attorney could draw an analogy between A and P in terms of the shared factors f1, f2 and f4. In other words the attorney would argue that where, as in A, factors f1 and f2 favor the plaintiff, the plaintiff in P should win as it did in case A, even though here, as there, factor f4 favors the defendant. The shared factors are relevant similarities between A and P.

An attorney arguing for the defendant would immediately see a way to respond to the plaintiff's argument citing case A. Case A is not exactly the same as problem P; P has two factors that case A does not: factor f3 which favors the plaintiff, and factor f5 which does not. The defendant's attorney would respond to the argument by distinguishing case A. She would dispute that case A justifies an outcome for plaintiff in P because in P, the extra factor f5 favors the defendant, making the defendant's position in P stronger than it was in A. From the defendant's point of view, that unshared factor f5 is a relevant distinction between A and P.

The argument context, in particular which side is responding to an opponent's case, helps to determine if a difference is a distinction. In this partially matched case, it determines which of the two factors is so important a difference as to

distinguish the case. The defendant's attorney could not claim that f3 was a relevant distinction between *A* and *P*. Factor f3 favors the plaintiff's side of the argument. It makes the plaintiff's position stronger in *P* than it was in *A* and is a feature that a plaintiff's attorney would emphasize as an extra reason for a favorable decision for plaintiff.

Distinctions can also help choose among partially matched cases. Cases *D* and *E* in Figure 2 each partially match *P*. Each has one factor that *P* does not have. Each was won by the plaintiff. Case *D* is the better case for the plaintiff to cite however. Case *D* is not distinguishable from *P*. If plaintiff cited *E*, the defendant would point out that f8 makes *E* a stronger case for the defendant than *P*. No such distinction applies to case *D*. Factor f5 actually helps the plaintiff; it makes case *D* stronger for the defendant than *P*, and yet the plaintiff still won.

An attorney's goal in selecting cases to cite is to select a case that is both favorable to his side and which has a maximum of factors in common with the problem situation. In Figure 1, case *B* would be a better case to cite for the plaintiff than *A*. Plaintiff won case *B* ($O(B) = +$) and the set of factors *A* shares with *P* is but a subset of those that *B* shares. In other words, *B* is more on point relative to *P* than *A*. The more on point a case, the better it is for a side to cite, assuming that side won the case. Note that determining how on point cases are involves comparing them in terms of set inclusion, not counting how many factors they share with the problem. The on pointness comparison is a partial order. A case *F* consisting of factors f2, f3 and f5 is neither less on point nor more on point than case *B*, but simply incomparable according to this definition. The fact that *B* shares four factors in common with *P*, while *F* shares three, does not make *B* more on point than *F*.

If the opponent won a more on point case, then that case is the basis of a potent response to an argument, a trumping counter-example. Suppose that in Figure 1, the

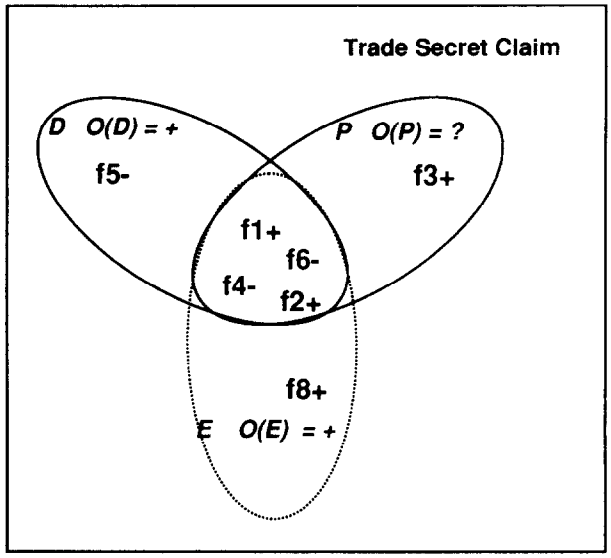

FIGURE 2. Selecting among partially matched cases.

defendant had won case B (i.e. $O(B) = -$). Case B would then be both more on point than case A and opposite in outcome. If plaintiff's attorney cited A, the opponent would cite B as a counter-example to A, one that effectively trumps plaintiff's point. Defendant's attorney would argue that A does not justify a decision for the plaintiff when there is a case B that has more in common with P (i.e. the extra shared pro-defendant factor f5) and was won by defendant.

To summarize, the Venn diagrams illustrate intuitively how precedents provide a means for justifying resolutions of competing factors and one metric for comparing cases: the overlap of factors relative to a problem. They also suggest that the context of the argument, including problem facts, cases available and the side arguing, can be used to assess symbolically the closeness of match and the factors that are salient in comparing the cases for purposes of the argument. In HYPO, relevant similarities and differences are defined in terms of factors. HYPO selects most on point cases to cite in arguments about a problem and cites various kinds of counter-examples, including trumping counter-examples, when making responses.

Factors also enable a different kind of comparison of cases that cannot be illustrated in the context of the Venn diagrams: comparing two cases in terms of the magnitude of the factor in each case. A factor's magnitude is a measure of how extreme the factor is in a case to which it applies. Suppose factor f2 deals with security measures; a plaintiff is better off to the extent that it adopted more security measures. In two cases, plaintiffs may have adopted security measures, but the plaintiff in case A may have adopted many more kinds of security measures than in case B. In other words, the magnitude of factor f2 is greater in case A than in case B.

Factors' magnitudes can often be represented in terms of numbers, ordered sets and partially ordered sets; comparisons of a factor's magnitudes in two cases can be computed by a corresponding definition of greater or less than. In the case of HYPO's representation of the security measures factor, magnitudes are compared in terms of whether one case's set of possible security measures includes the other's.

A factor's magnitude should be distinguished from its significance or weight in an argument. A factor's weight, as discussed above, cannot be represented with a number or static hierarchy. Nevertheless, a factor's magnitude may affect argument weight. Although factor f2 weighs in favor of the plaintiff in case A, it would have even more weight if its magnitude were greater in case A. Factor f2 may have a higher magnitude in case B than in case A. All other things being equal, the difference in magnitude may make case A a better case to cite for plaintiff where the problem's plaintiff also adopted only minimal security measures. In terms of factor magnitudes, Case A is better because the defendant would distinguish case B, pointing out that it involved more security measures than the problem's plaintiff undertook.

Neither do the Venn diagrams illustrate another aspect of reasoning with factors and cases. It often makes sense to speak of a factor as *almost* applying to a case or, in other words, as a near miss with respect to a case. If a factor could apply to a problem with only a small change in that problem's facts, then cases that have that factor are almost relevant. In Section 6.2, I define a sense in which a factor is a near miss.

4. A legal example of arguing with factors and cases

To make the concept of arguing with factors and cases more concrete, and to introduce an extended example of how HYPO reasons with cases, consider a specific legal problem like the following trade secrets dispute.

> Amexxco, a major oil company, complains about the actions of its former employee named G. Whiz. While working for Amexxco, Mr. Whiz had developed a computer program, called Dipper, for analysing drilling logs of oil wells. Although on the Amexxco payroll since 1980, Whiz developed the program on his own initiative and without Amexxco's support. In fact, for four years, Amexxco had repeatedly directed Whiz to drop the Dipper in favor of another approach. Then in a trial experiment conducted by Whiz, the Dipper discovered a major oil well. A year ago in 1987, in a salary dispute, Whiz quit Amexxco's employ and entered into an employment contract with Amexxco's competitor, Exxssinc, to work on computerized analysis of oil drilling logs. Whiz had signed a nondisclosure agreement with Amexxco in which he undertook to maintain confidentiality with respect to all of Amexxco's trade secrets. Amexxco wants to know what legal rights it has against its former employee, Whiz, and against Exxssinc.

An attorney familiar with trade secret law and cases would recognize that Amexxco has a possible legal claim of trade secret misappropriation. Probably she or he would also recognize various stereotypical collections of facts in the problem i.e. factors, some of which strengthen Amexxco's claim and some of which weaken it. On the one hand, two factors strengthen Amexxco's claim:

(1) f1: Amexxco adopted some (at least minimal) measures to protect its trade secrets, it appears to have secured a nondisclosure agreement from at least one employee.
(2) f2: it secured a nondisclosure agreement from the particular person, here the former employee, who arguably has taken its trade secrets in the Dipper program.

On the other hand, two other factors weaken Amexxco's claim:

(1) f3: the employee was the sole developer of the Dipper program, i.e. he performed the work on his own initiative and without Amexxco's support.
(2) f4: although the employee had signed the nondisclosure agreement, the agreement did not specifically refer to the Dipper program.

Having analysed the strengths and weaknesses of Amexxco's claim in terms of the competing factors, the question is how should the competing factors be resolved? There are two possible solutions, either the plaintiff has a valid claim for trade secret misappropriation or it does not. The question may arise in different contexts. At the trial court level, the judge or jury decides the facts that have been proven and, inferentially, the factors that apply to a case. The judge may entertain legal arguments from the disputing parties' attorneys on the issue of whether the factors are sufficient, or if the factors compete, how to resolve them in determining the victor or in fashioning a charge (i.e. instructions) to the jury. If the dispute has reached an appeals court, an appeals judge would entertain similar legal arguments in assessing the correctness of the decision below.

For the attorneys representing the competing parties, the questions are: what justifications can they provide in their legal arguments for resolving the competing

factors in favor of their side? What justifications can they expect the opposing attorney to make in support of resolving the competing factors in favor of the other side? Attorneys need to consider these questions, not only in putting arguments to a judge but in assessing whether they should commence a lawsuit in the first place. In assessing the client's prospects, they need to consider, hypothetically, the arguments justifying a favorable resolution of competing factors that they or their opponents are likely to be able to prove to a jury.

Arguing in terms of factors alone, is not a sufficient justification for a legal argument. Instead, advocates justify a resolution symbolically by citing a similar past case or precedent, one which involved some or all of the same competing factors. Amexxco's attorney, for example, could not simply argue that Amexxco should win because factors f1 and f2 favor its claim. The defendants would respond that factors f3 and f4 support their position. The argument is a stand-off, not just because there are two factors on each side; it would be a stand-off even if one side could cite more factors than the other. It is a stand-off because neither attorney has cited a legal authority such as a precedent for the conclusion that the competing factors should be resolved in their favor. If there were an authoritative scheme in law which assigned weights to the factors, an attorney could cite the scheme, combine the weights and resolve the conflict. There is, however, no such scheme. The immediate goal of the attorneys then is to find favorably decided past cases that have maximal overlaps of factors in common with the problem.

5. Brief overview of HYPO and its outputs

HYPO assists attorneys by finding such cases and demonstrating how they can be used in arguments *pro* and *contra* a side's position in the problem. Here, in a brief overview, is how HYPO works. HYPO has 30 trade secrets and related cases in its database of cases, its Case Knowledge Base (CKB). A Case Representation Language has been designed for representing legal cases and problem situations at two levels of abstraction. Legal Case Frames are employed to represent the detailed, lower level facts of a case. In addition, a second language, *Factual Predictes* summarizes in a slightly more abstract representation the lower level case facts represented in Legal Case Frames.

HYPO also has a library of 13 *Dimensions*. Dimensions are a knowledge representation construct for representing and reasoning with factors. Each of HYPO's Dimensions represents a factor relevant for trade secrets or related legal claims. The *Amexxco* problem's four factors correspond to the following Dimensions:

Factor	Dimension	Favors
f1	*Security-Measures*	Plaintiff
f2	*Agreed-Not-To-Disclose*	Plaintiff
f3	*Employee-Sole-Developer*	Defendant
f4	*Nondisclosure-Agreement-Specific*	Defendant

Each Dimension has prerequisites with which HYPO can determine whether the Dimension applies to a case or problem. The prerequisites are defined in terms of Factual Predicates. Saying that a factor applies to a problem means that the

corresponding Dimension's prerequisites are satisfied by the problem's Factual Predicates. Each case in the CKB is indexed by the Dimensions that HYPO previously found applicable.

From this point on, instead of factors, I will speak in terms of Dimensions. Each of HYPO's Dimensions, it should be remembered, represents a factor. All of HYPO's Dimensions are supplied by the designer, not generated by the program.

HYPO begins with a problem like the *Amexxco* case already represented by the user in the Legal Case Frames. A case presented to HYPO for analysis is referred to as the current fact situation (cfs). A case editor provides menu-driven assistance in filling out the problem representation. HYPO has no capability for analysing natural language descriptions of the problem.

Once the problem is presented, HYPO analyses it to determine the Factual Predicates and the Dimensions that apply to the problem. In effect, it determines the set of Dimensions that covers the problem. Then HYPO retrieves all of the cases that are indexed by any Dimension that applies to the problem. These are the relevant cases: each shares at least one factor with the problem. HYPO then organizes the relevant cases so that it can select the ones that are most on point to the problem situation. From the most on point cases, it computes the best cases for each side to cite, distinctions between those cases and the problem, and counter-examples to those cases. In making these selections, HYPO performs set operations like those illustrated in the Venn diagrams of Figures 1 and 2. These operations will be discussed in much more detail.

Having identified cases, counter-examples and distinctions, HYPO generates arguments, called 3-Ply Arguments, to demonstrate how the cases can be used and responded to by each side. The arguments present opposing viewpoints of both sides of the dispute, plaintiff and defendant. They are called "3-Ply" because they step through three turns, or plies, of an argument citing a precedent. HYPO makes a point for one side by drawing an analogy between the problem and precedent, responds to the point by distinguishing the precedent or citing counter-examples, and rebuts the response by, for example, distinguishing the counter-example.

For each of the best cases for each side to cite, HYPO outputs a 3-Ply Argument. For the *Amexxco* problem where plaintiff and defendant each have one best case, HYPO automatically outputs two 3-Ply Arguments.

[1]
⇒ Point for Plaintiff as Side-1:
Where: Plaintiff and defendant entered into a nondisclosure agreement. Plaintiff adopted security measures.
Even though: Employee defendant was sole developer of plaintiff's product. The nondisclosure agreement did not specifically refer to plaintiff's product.
Plaintiff should win a claim for Trade Secrets Misappropriation.
Cite: *Structural Dynamics Research Corp. v. Engineering Mechanics Research Corp.* 401 F. Supp. 1102 (E.D. Mich. 1975).

⇐ Response for Defendant as Side-2:
Structural Dynamics Research Corp. v. Engineering Mechanics Research Corp. is distinguishable because:
In Structural Dynamics, plaintiff adopted more security measures than in Amexxco.
In Structural Dynamics, plaintiff's former employee brought product development information to defendant. Not so in Amexxco.

Counter-examples:
Amoco Production Co. v. Lindley 609 P.2d 733 (Okla. 1980), is as on point and held for defendant.

⇒ Rebuttal for Plaintiff as Side-1: None.

[2]
⇒ Point for Defendant as Side-1;
Where: Employee defendant was sole developer of plaintiff's product. The nondisclosure agreement did not specifically refer to plaintiff's product.
Even though:
Plaintiff and defendant entered into a nondisclosure agreement. Plaintiff adopted security measures.
Defendant should win a claim for Trade Secrets Misappropriation.
Cite: *Amoco Production Co. v. Lindley* 609 P.2d 733 (Okla. 1980).

⇐ Response for Plaintiff as Side-2:
Counter-examples:
Structural Dynamics Research Corp. v. Engineering Mechanics Research Corp. 401 F. Supp. 1102 (E.D. Mich. 1975), is as on point and held for plaintiff.

⇒ Rebuttal for Defendant as Side-1:
Structural Dynamics Research Corp. v. Engineering Mechanics Research Corp. is distinguishable because:
In Structural Dynamics, plaintiff adopted more security measures than in Amexxco.
In Structural Dynamics, plaintiff's former employee brought product development information to defendant. Not so in Amexxco.
Note: Plaintiff's response would be strengthened if:
Plaintiff's former employee brought product development information to defendant.
Cf. Structural Dynamics Research Corp. v. Engineering Mechanics Research Corp. 401 F. Supp. 1102 (E.D. Mich. 1975)

HYPO invites the user to select cases manually to try out in arguments for either side. HYPO's graphic outputs depicting the relevance of cases from the CKB to the problem (i.e. Claim Lattices, described below) guide the user as to which cases would be interesting to try out. Here, HYPO generates three arguments for user-selected cases.

[3]
⇒ Point for Plaintiff as Side-1:
Where: Plaintiff and defendant entered into a nondisclosure agreement. Plaintiff adopted security measures.
Plaintiff should win a claim for Trade Secrets Misappropriation.
Cite: *Eastern Marble Products Corp. v. Roman Marble Inc.* 364 N.E.2d 799 (Mass. 1977).

⇐ Response for Defendant as Side-2:
Eastern Marble Products Corp. v. Roman Marble, Inc. is distinguishable because:
In Eastern-Marble, plaintiff adopted more security measures than in Amexxco.
In Amexxco, employee defendant was sole developer of plaintiff's product. Not so in Eastern-Marble.
Counter-examples:
Amoco Production Co. v. Lindley 609 P.2d 733 (Okla, 1980), is more on point and held for defendant where it was also the case that: the nondisclosure agreement did

not specifically refer to plaintiff's product. Employee defendant was sole developer of plaintiff's product.

⇒ Rebuttal for Plaintiff as Side-1: None.

[4]
⇒ Point for Defendant as Side-1:
Where: Employee defendant was sole developer of plaintiff's product.
Defendant should win a claim for Trade Secrets Misappropriation.

Cite: *Wexler v. Greenberg* 160 A.2d 430 (Sup. Ct. Pa 1960).

⇐ Response for Plaintiff as Side-2:
Wexler v. Greenberg is distinguishable because:
In Amexxco, plaintiff adopted security measures. Not so in Wexler. In Amexxco, plaintiff and defendant entered into a nondisclosure agreement. Not so in Wexler.
Counter-examples:
Structural Dynamics Research Corp. v. Engineering Mechanics Research Corp. 401 F.Supp. 1102 (E.D. Mich. 1975), is more on point and held for plaintiff where it was also the case that: Plaintiff adopted security measures. The nondisclosure agreement did not specifically refer to plaintiff's product. Plaintiff and defendant entered into a nondisclosure agreement.

⇒ Rebuttal for Defendant as Side-1:
Structural Dynamics Research Corp. v. Engineering Mechanics Research Corp. is distinguishable because:
In Structural Dynamics, plaintiff adopted more secuirty measures than in Amexxco.
In Structural Dynamics, plaintiff's former employee brought product development information to defendant. Not so in Amexxco.

[5]
⇒ Point for Defendant as Side-1:
Where: the nondisclosure agreement did not specifically refer to plaintiff's product.
Even though: Plaintiff and defendant entered into a nondisclosure agreement.
Defendant should win a claim for Trade Secrets Misappropriation.
Cite: *Motorola, Inc. v. Fairchild Camera and Instrument Corp.* 366 F.Supp. 1173 (D. Arizona 1973).

⇐ Response for Plaintiff as Side-2:
Motorola, Inc. v. Fairchild Camera and Instrument Corp. is distinguishable because:
In Amexxco, Plaintiff adopted security measures. Not so in Motorola.
Counter-examples:
Structural Dynamics Research Corp. v. Engineering Mechanics Research Corp. 401 F. Supp. 1102 (E.D. Mich. 1975), is more on point and held for plaintiff where it was also the case that: Plaintiff adopted security measures. Employee defendant was sole developer of plaintiff's product.

⇒ Rebuttal for Defendant as Side-1:
Structural Dynamics Research Corp. v. Engineering Mechanics Research Corp. is distinguishable because:
In Structural Dynamics, plaintiff adopted more security measures than in Amexxco.
In Structural Dynamics, plaintiff's former employee brought product development information to defendant. Not so in Amexxco.

In its 3-Ply Arguments, HYPO makes a symbolic comparison between a problem and the most relevant precedents in terms of Dimensions. The points in each of the first two arguments cite a case, such as *Structural Dynamics* or *Amoco,* that has some maximal set of Dimensions in common with the problem. The analogy between the case and the problem (prefaced by the terms "Where" and "Even though") recite the shared Dimensions and indicate which side the Dimension favors. "Even though" prefaces Dimensions that favor the opponent of the side making the point. The responses distinguish the cases by pointing out Dimensions that are not shared by the precedent or problem. Also they distinguish cases in terms of magnitude. Where possible, the response also cites a counter-example. In Argument [1], for example, HYPO points out that the *Amoco* case is as on point as, and contrary to, *Structural Dynamics.* In Argument [3], it points out that *Amoco* is *more* on point than and contrary to *Eastern Marble.* In other words, *Amoco* is a trumping counter-example.

In addition to outputting 3-Ply Arguments, HYPO has two other kinds of output. First it performs a rudimentary evaluation of the relative strengths of the two side's arguments which is presented in a case summary. HYPO determines that a side has a stronger argument if it is the only side where not all of its points can be trumped, i.e. not all of its best cases have trumping counter-examples. If both sides have points that are not trumped, HYPO does not attempt to further evaluate the argument. It concludes that each side has a strong argument and presents each side's best cases. That is the situation in the *Amexxco* problem for which HYPO generates this summary:

> On a claim for Trade Secrets Misappropriation, both sides can make a strong argument.
> Plaintiff can cite the following cases for which there are no more on point counter-examples:
> Structural Dynamics Research Corp. v. Engineering Mechanics Research Corp. 401 F. Supp. 1102 (E.D. Mich. 1975).
> Defendant can cite the following cases for which there are no more on point counter-examples:
> Amoco Production Co v. Lindley 609 P.2d 733 (Okla. 1980)

HYPO also outputs a list of suggested hypothetical modifications of the problem situation that would strengthen or weaken the plaintiff's side. Depending which side the suggested modification favors, the plaintiff could make additional points by citing new best cases or its existing points could be trumped by new counter-examples. Such suggestions are useful in preparation for trial because an attorney cannot assume that she or he knows or can prove all of the facts that favor her side, or that her opponent will not prove some unfavorable ones. The hypotheticals suggest new facts that she or he may elict from her client or prepare to rebut. HYPO can also pose hypotheticals by pushing a competing Dimension in a case to extremes. An advocate could use such hypotheticals to convince an appellate court that a factor is or is not very important. For the *Amexxco* problem, HYPO suggests the following hypothetical modifications:

> Hypotheticals to consider on claim of Trade Secrets Misappropriation in fact

situation of Amexxco:

Plaintiff's position would be strengthened in following situations:
Suppose:
Defendant's access to plaintiff's product information saved it time or expense.
Plaintiff's former employee brought product development information to defendant.
Cf. Analogic Corp. v. Data Translation, Inc. 358 N.E.2d 804 (S.J.C. Mass. 1976)

Defendant's access to plaintiff's product information saved it time or expense.
Defendant paid plaintiff's former employee to switch employment. Plaintiff's former employee brought product development information to defendant.
Cf. Space Aero Products Co. v. R.E. Darling Co. 208 A.2d 74 (Ct. App. Md. 1965)

Defendant's access to plaintiff's product information saved it time or expense.
Defendant paid plaintiff's former employee to switch employment.
Cf. Telex Corp. v. IBM Corp. (1) 510 F.2d 894 (10 Cir. 1975)

Plaintiff's former employee brought product development information to defendant.
Cf. Structural Dynamics Research Corp. v. Engineering Mechanics Research Corp. 401 F. Suppl 1102 (E.D. Mich. 1975)

Defandant's position would be strengthened in following situations:
Suppose:
Plaintiff disclosed its product information in negotiations with defendant. Plaintiff disclosed its product information to outsiders.
Cf. Crown Industries Inc. v. Kawneer Co. 335 F. Supp. 749 (N.D. Ill. 1971)

Plaintiff disclosed its product information to outsiders.
Cf. Midland-Ross Corp. v. Yokana 293 F.2d 411 (3 Cir. 1961)

Plaintiff disclosed its product information in negotiations with defendant. Plaintiff's product information was about customer business relations.
Cf. Automated-Systems v. Service-Bureau 401 F.2d 619 (10 Cir. 1968)

Plaintiff disclosed its product information to outsiders.
Cf. Midland-Ross Corp. v. Sunbeam Equipment Corp. 316 F. Supp. 171 (W.D. Pa. 1970)

HYPO computes suggested hypotheticals by using set operations similar to those illustrated in the Venn diagrams. Each hypothetical is based on a potentially more on point case or counter-example, as defined below. If the user wishes, HYPO will instantiate the suggested hypothetical modifications (that is, create a hypothetically modified version of the problem), generate new 3-Ply Arguments for the modified problem and compare the arguments with those for the old problem version.

6. Knowledge representation in HYPO

The overview discussion has mentioned a number of knowledge sources employed in HYPO including a Case Representation Language, Dimensions for representing factors, the Case Knowledge Base, and standards for evaluating arguments. The knowledge sources are defined more completely here and illustrated in the context of the *Amexxco* example. The main argument evaluation standards, which are represented procedurally in HYPO, are discussed in Section 7.6.

6.1. CASE REPRESENTATION LANGUAGE

The Case Representation Language is comprised of a two-tiered, frame-based language for representing legal cases and problem situations. The first tier, Legal Case Frames, represents detailed facts involved in the case, and if the case has already been decided, certain aspects of the judicial decision. The second tier, Factual Predicates, summarizes the lower level facts represented in the Legal Case Frames.

6.1.1. Legal Case Frames

Each case in the CKB, problem situation posed to HYPO, or hypothetical posed by HYPO, has a top-level Legal Case Frame and various underlying sub-frames. The top-level frame contains the case name, citation information, parties to the dispute and their roles as plaintiff or defendant, type of legal claims raised and the decision of the court. For problem situations and hypotheticals, the top-level frame is mostly blank because those cases have not been decided by a court. Here, for example, is the top-level case frame for the *Amexxco* problem.

Top-level Legal Case Frame for *Amexxco*

CASE: Amexxco Production Co. v. Gwhiz

SHORT-TITLE: Amexxco

CITATION: hypothetical case

DATE: 1987

PARTY-LIST: (Corporate-Party:Amexxco
Corporate-Party:Exxssinc
Employee-Party:Gwhiz)

ROLE-PARTY-ALIST:
((Plaintiff Corporate-Party:Amexxco)
(Defendant Corporate-Party:Exxssinc Employee-Party:Gwhiz))

DECISION-FOR: Nil

CLAIMS-HELD-FOR: Nil

CLAIM-OR-DEFENSE-LIST: Nil

DIMENSIONS-LIST: Nil

CASES-CITED: Nil

In illustrating frames, the following syntax is employed. The names of slots apear in small capitals followed by a colon (e.g. PARTY-LIST:). The value of the slot is whatever appears after the space following the colon. Often, other instantiated frames from the case appear in a slot value. For example, the Corporate-Party frame whose name is Amexxco appears in the value of PARTY-LIST; it is written as "Corporate-Party:Amexxco". Slot values may be lists like the value of the PARTY-LIST slot above. They may also be association lists, like the value of the ROLE-PARTY-ALIST above which indicates that the plaintiff is the Corporate-Party:Amexxco and the defendants are the Corporate-Party:Exxssinc and Employee-Party:Gwhiz.

The underlying frames represent detailed factual objects, relations and events that are typically present in trade secrets disputes. As shown in the following complete list, the frames represent parties to the lawsuit, the parties' products, alleged confidential information, employment relations involving parties and products and various contracts. For any particular case, its Legal Case Frames are nested (i.e. some slot values are pointers to other case frames.)

Parties to lawsuit:
- Party-To-Lawsuit
- Person-Party
- Corporate-Party
- Employee-Party
- Person
- Corporation
- Employee

Product-related:
- Product
- Intrinsic-Similarities

Alleged confidential information:
- Knowledge
- Security-Breach
- Disclosure-Event

Employment relations involving parties and products:
- Employment
- Employment-Change
- Product-Worked-On

Contracts:
- Agreement
- Noncompetition-Covenant
  Nondisclosure-Agreement
- Promise
- Reliance

The easiest way to describe the underlying Legal Case Frames is to focus on parts of the case frame representation of the *Amexxco* case. Like a number of cases in the CKB, the *Amexxco* problem involved a "common employee scenario" where an employee allegedly brought confidential information from the plaintiff to the defendant. In the *Amexxco* fact situation, a former employee brought secrets he developed himself to a competing employer. Here are the underlying Legal Case Frames for representing this scenario. The Employee-party frame shows that G. Whiz was employed by both Amexxco and Exxssinc, the two corporate parties in the suit.

Employee-Party:Gwhiz

NAME: GWhiz

CASE: Amexxco Production Co. v. Gwhiz

CASE ROLE: Defendant

EMPLOYMENTS:
Employment:Of-Gwhiz-By-Amexxco
Employment:Of-Gwhiz-By-Exxssinc

EMPLOYERS:
Corporate-Party:Amexxco
Corporate-Party:Exxssinc

Details of G. Whiz's employment by each party are shown in the Employment frames. The dates of employment show that G. Whiz started working for Amexxco in 1980 and then left in 1987. While employed by Amexxco, he worked on the Dipper product. He started at Exxssinc in 1987. While at Exxssinc, he worked on Exxssinc's competing system. There is no information about Whiz's having signed a non-competition agreement.

Employment:Of-Gwhiz-By-Amexxco

CASE: Amexxco Production Co. v Gwhiz

EMPLOYEE: Employee-Party:Gwhiz

EMPLOYER: Corporate-Party:Amexxco

STARTING DATE: 1980

ENDING DATE: 1987

THERE IS COVENANT NOT TO COMPETE: Nil

THERE IS CHANGE OF EMPLOYERS:
Employment-Change:Of-Gwhiz-From-Amexxco-To-Exxssinc

PRODUCT WORKED ON:
Product-Worked-On:Work-On-Dipper

The frame representing the Employment Change indicates that there is no information in the case that G. Whiz received anything of value for switching employers or that he brought any records, devices, or code with him. Note that each slot contains a "Nil", indicating an absence of information.

Employment-Change:Of-Gwhiz-From-Amexxco-To-Exxssinc

CASE: Amexxco Production Co. v. Gwhiz

FORMER EMPLOYER: Corporate-Party:Amexxco

FORMER EMPLOYMENT: Employment:Of-Gwhiz-By-Amexxco

VALUE RECEIVED TO MAKE CHANGE: Nil

RECORDS, DEVICES, CODE BROUGHT BY EMPLOYEE: Nil

The employee's relationship to the products he worked on for each employer are represented in a Product frame for the Dipper, two Product-Worked-On frames, Work-On-Dipper and Work-On-Exxssinc-System (not shown). They indicate that G. Whiz used the same information, Knowledge:About-Dipper on each product. The Work-On-Dipper frame also indicates that Whiz was the sole developer of the Dipper product.

Product:Dipper

CASE: Amexxco Production Co. v. Gwhiz

A KIND OF: Well-Log-analyzer

PRODUCT USED FOR: analysing oil wells

GENERAL PRODUCT MARKET: Oil Companies

COMPETITORS PRODUCT ALIST: (Corporate-Party:Exxssinc Product:Exxssinc-System)

DEVELOPER: Corporate-Party:Amexxco

EMPLOYMENT WORKED ON PROJECT LIST: Employment:Of-Gwhiz-By-Amexxco

PROJECT DEVELOPMENT START: Nil

PROJECT DEVELOPMENT END: Nil

PROJECT DEVELOPMENT TIME: Nil

EXPENDITURES MADE: Nil

KNOWLEDGE USED: Knowledge:About-Dipper

SECURITY MEASURES LIST: (Minimal-Measures)

SECURITY BREACH LIST: Nil

Product-Worked-On:Work-On-Dipper

CASE: Amexxco Product Co. v. Gwhiz

PRODUCT: Product:Dipper

EMPLOYMENT: Employment:Of-Gwhiz-By-Amexxco

KNOWLEDGE EMPLOYED: Knowledge:About-Dipper

EMPLOYEE ROLE IN PRODUCTION DEVELOPMENT: Sole Developer

The knowledge used in connection with the Dipper product is represented in a Knowledge frame.

Knowledge:About-Dipper

CASE: Amexxco Production Co. v. Gwhiz

NAME: About-Dipper

KNOWLEDGE ABOUT: Product:Dipper

KIND OF KNOWLEDGE: Technical

PARTY WITH ACCESS LIST:

(Corporate-Party:Amexxco Corporate-Party:Exxssinc Employee-Party:Gwhiz)

OTHER PERSON WITH ACCESS LIST: Nil

NUMBER PERSONS WITH ACCESS: Nil

SPECIFIC DISCLOSURE EVENTS:

(Disclosure-Event:Amexxco-Discloses-To-Exxssinc-Via-Gwhiz)

NUMBER DISCLOSEES: Nil

PERCENT DISCLOSEES RESTRICTED: Nil

GENERALLY KNOWN IN INDUSTRY: Nil

The employee's switch of employers, work on competing products and use of the same knowledge for each one constitutes a disclosure event and is represented by its own frame. As the frame indicates, the disclosure event was subject to a restriction, a nondisclosure agreement.

Disclosure-Event:Amexxco-Discloses-To-Exxssinc-Via-Gwhiz

CASE: Amexxco Production Co. v. Gwhiz

INFO DISCLOSED: Knowledge:About-Dipper

DISCLOSER: Corporate-Party:Amexxco

DISCLOSEE: Corporate-Party:Exxssinc

HOW DISCLOSURES MADE:
Employment-Change:Of-Gwhiz-From-Amexxco-To-Exxssinc

TIME TO GENERATE INFO FROM DISCLOSURE: Nil

NONDISCLOSURE AGREEMENTS:

Nondisclosure-Agreement:Between-Gwhiz-And-Amexxco-2

Finally, the frame representing the nondisclosure agreement between G. Whiz and Amexxco indicates that both sides received some consideration and that the agreement did not specifically refer to the Dipper product.

Nondisclosure-Agreement:Between-Gwhiz-And-Amexxco-2

CASE: Amexxco Production Co. v. Gwhiz

DISCLOSER: Corporate-Party:Amexxco

DISCLOSEE: Employee-Party:Gwhiz

CONSIDERATION RECEIVED BY PARTY:
Corporate-Party:Amexxco got promise
Employee-Party:Gwhiz got employment

DID AGREEMENT REFER TO PRODUCT: Negative

DURATION OF PROHIBITION: Nil

EXPRESS OR IMPLIED: Express

DATE ENTERED: Nil

6.1.2. Factual Predicates

A language of Factual Predicates has been invented to summarize lower level case facts represented in Legal Case Frames. They are generalized factual statements that confirm whether certain legally significant relationships are true in the case. HYPO infers whether a Factual Predicate is satisfied by using a retrieval method associated with the predicate. The retrieval method tests information about the case contained in the Legal Case Frames. For example, there are Factual Predicates to test whether there is a corporate plaintiff who makes a product that has associated development information; whether there is an employee who switched from working for plaintiff to working for defendant; whether the defendant entered into a

nondisclosure agreement with plaintiff; or whether the defendant saved product development expense relative to plaintiff.

For any case or problem, Factual Predicates are recorded in a data structure called an Interpretation Frame. Each Interpretation Frame slot corresponds to a Factual Predicate. Each slot value corresponds to the value of the Factual Predicate in a case or problem. Factual Predicates may have the following values: "Nil" indicating no information, "Negative" indicating false, "Affirmative" indicating true, a name of a specific instantiated underlying case frame, or a slot value from an instantiated case frame. The case frame names and slot values stand for affirmative answers and provide additional information; they mean that the Factual Predicate's value is the instantiated case frame itself, or was computed from information contained in the frame's slots. Every case is analysed using an identical blank Interpretation Frame; the slot values of the filled in Interpretation Frame differ from case to case. The common employee scenario of the *Amexxco* problem calls into play a set of Factual Predicates illustrated below. Here are excerpts of the completed Interpretation Frame for *Amexxco*:

Excepts from Interpretation Frame for *Amexxco*

CASE: Amexxco Production Co. v. Gwhiz

THERE IS A CORPORATE PLAINTIFF:
Corporate-party:Amexxco

THERE IS A CORPORATE DEFENDANT:
Corporate-Party:Exxssinc

THERE IS AN EMPLOYEE DEFENDANT:
Employee-Party:Gwhiz

PLAINTIFF MAKES A PRODUCT:
Product:Dipper

PLAINTIFF HAS PRODUCT INFORMATION:
Knowledge:About-Dipper

DEFENDANT MAKES A PRODUCT:
Product:Exxssinc-System

PLAINTIFF AND DEFENDANT COMPETE:
Affirmative

EMPLOYEE WORKED FOR PLANTIFF:
Employment:Of-Gwhiz-By-Amexxco

EMPLOYEE WORKED FOR DEFENDANT:
Employment:Of-Gwhiz-By-Exxssinc

EMPLOYEE WORKED FOR BOTH PLANTIFF AND DEFENDANT: Affirmative

EMPLOYEE SWITCHED FROM WORKING FOR PLANTIFF TO WORKING FOR DEFENDANT:
Employment-Change:Of-Gwhiz-From-Amexxco-To-Exxssinc

EMPLOYEE RECEIVED SOMETHING OF VALUE TO SWITCH EMPLOYMENT: Nil

EMPLOYEE BROUGHT PLAINTIFF'S PRODUCT DEVELOPMENT TOOLS TO DEFENDANT: Nil

EMPLOYEE WORKED ON PLAINTIFF'S PRODUCT:
Product-Worked-On:Work-on-Dipper

DEFFENDANT HAD ACCESS TO PLANTIFF'S PRODUCT VIA EMPLOYEE:
Disclosure-Event:Amexxco-Discloses-To-Exxssinc-Via-Gwhiz

DEFFENDANT OR EMPLOYEE ENTERED INTO NONDISCLOSURE AGREEMENT WITH PLAINTIFF:
Nondisclosure-Agreement:Between-Gwhiz-and-Amexxco-2

PLAINTIFF MADE SOME DISCLOSURES TO OUTSIDERS:
Nil

TYPE OF PRODUCT INFORMATION IS TECHNICAL:
Nil

DEFENDANT SAVED PRODUCT DEVELOPMENT EXPENSE RELATIVE TO PLAINTIFF:
Negative

PLAINTIFF ADOPTED SECURITY MEASURES:
Minimal-measures

PLAINTIFF DISCLOSED PRODUCT INFORMATION TO DEFENDANT IN NEGOTIATIONS:
Nil

EMPLOYEE ENTERED INTO NONCOMPETITION AGREEMENT WITH PLAINTIFF:
Nil

NONDISCLOSURE AGREEMENT SPECIFICALLY COVERED PLAINTIFF'S PRODUCT:
Negative

EMPLOYEE WAS SOLE DEVELOPER OF PLAINTIFF'S PRODUCT:
Sole-Developer

By reading the Interpretation Frames's slots one can summarize the facts of the case. For example, this Interpretation Frame summarizes the facts of *Amexxco*, which fit into the common employee scenario, that the employee switched from working for plaintiff to working for defendant and had worked on plaintiff's Dipper product and thus was a route by which defendant gained access to information about the product. It also reports that there was no information that the employee received something of value to switch or brought plaintiff's product development tools. The employee however, had entered into a nondisclosure agreement with plaintiff.

6.2. DIMENSIONS

HYPO's 13 Dimensions correspond to 13 factors that affect trade secrets and related claims. They are drawn from the Restatement (First) of Torts, Section 757, and various other legal treatises and law review articles for trade secret law (such as Gilburne & Johnston, 1982; Milgrim, 1985; Nimmer, 1985). Authors of such works commonly group together cases that share some important factor. Frequently, the author includes cases won by defendants as well as plaintiffs and cases that show a range of magnitudes of the factor or a variety of additional counterveiling factors. The nature of the factual comparisons is usually evident from the author's comments or those of the cases reported. The Dimensions are:

(1) *Competitive-Advantage*—Plantiff strengthened the greater the competitive advantage gained by the defendant.
(2) *Vertical-Knowledge*—Plaintiff strengthened to the extent that the knowledge does not pertain to customer business methods.

(3) *Secrets-Disclosed-Outsiders*—Plaintiff strengthened the fewer disclosures to outsiders it has made of confidential information.
(4) *Outsider-Disclosures-Restricted*—Plaintiff strengthened to the extent that disclosees are restricted from disclosing confidential information to others.
(5) *Consideration*—Plaintiff strengthened to the extent that defendant received something of value for entering into contract.
(6) *Bribe-Employee*—Plaintiff strengthened the more money, stock, or other benefits the defendant gave to plaintiff's former employees to switch employment.
(7) *Noncompetition-Agreement*—Plaintiff strengthened to the extent that employee entered noncompetition agreement.
(8) *Brought-Tools*—Plaintiff strengthened to the extent that former employee brought product-related tools to defendant.
(9) *Agreed-Not-To-Disclose*—Plaintiff strengthened to the extent that defendant entered into a nondisclosure agreement.
(10) *Employee-Sole-Developer*—Plaintiff strengthened to the extent that defendant was not the sole developer of the confidential information.
(11) *Nondisclosure-Agreement-Specific*—Plaintiff strengthened to the extent that the nondisclosure agreement specifically referred to plaintiff's product.
(12) *Disclosure-In-Negotiations*—Plaintiff strengthened to extent that it did not disclose secret to defendant in negotiations.
(13) *Security-Measures*—Plaintiff strengthened the more security measures it took to protect its confidential information.

The *Security-Measures* Dimension, Figure 3, captures one factor that courts take into account in assessing the strength of a plaintiff's trade secrets claim: the plaintiff is in a stronger position to the extent that it has adopted more comprehensive security measures to protect the information. The *Secrets-Disclosed-Outsiders* Dimension, Figure 4, captures another factor: the plaintiff is in a weaker position to the extent that plaintiff made disclosures of the information to more outsiders.

Each Dimension is a structured object comprising prerequisites, focal slot, range, comparison type, pro-plaintiff direction and focal slot prerequisite. A Dimension's prerequisites are a list of Factual Predicates, all of which must be satisfied in a problem for the Dimension to be said to apply. To paraphrase the list of prerequisites of the *Security-Measures* Dimension (Figure 3), it requires that there be a corporate plaintiff, who makes a product, about which there is product development information, that there be a corporate defendant who makes a competing product, and that the plaintiff adopt some security measures. Since, as the *Amexxco* Interpretation Frame indicates, these prerequisites are satisfied, HYPO concludes that the *Security-Measures* Dimension applies.

The focal slot, range, comparison type and pro-plaintiff direction represent a Dimension's magnitude in a case. They allow cases to which a given Dimension applies to be compared in terms of the magnitude of the Dimension in each case. Focal slots single out the particular facts that make a case stronger or weaker along the Dimension in the sense that the case is a more or less extreme example. The focal slot indicates where to look in a case's underlying frame representation to find the facts that place the case somewhere along the Dimension. For example, the focal slot of the *Security-Measures* Dimension indicates that the value of the

SHORT-TITLE: Security-Measures

CLAIMS: Trade Secrets Misappropriation

PREREQUISITES:
- There is a corporate plaintiff
- There is a corporate defendant
- Plaintiff makes a product
- Plaintiff and defendant compete
- Plaintiff has product information
- Defendant makes a product
- Plaintiff's and defendant's products compete
- Plaintiff adopted security measures

FOCAL SLOT PREREQUISITE:
- Plaintiff adopted security measures

FOCAL SLOT:
- Plaintiff's Product: SECURITY-MEASURES-LIST

RANGE:
- Minimal-Measures
- Access to Premises Controlled
- Restrictions on Entry by Visitors
- Restrictions on Entry by Employees
- Product Marked Confidential
- Employee Trade Secret Program Exists
- Restrictions on Hardcopy Release
- Employee Nondisclosure Agreements

COMPARISON TYPE: More versus less

PRO PLAINTIFF DIRECTION: More

FIGURE 3. The *Security-Measures* Dimension

plaintiff's product's SECURITY-MEASURES-LIST slot places a case somewhere on that Dimension. The focal slot of the *Secrets-Disclosed-Outsiders* Dimension indicates that the NUMBER-DISCLOSEES slot of the plaintiff's product knowledge frame contains the relevant datum. The range information includes the range of possible values of the focal slots. The pro-plaintiff direction defines the direction in the range that strengthens the plaintiff's position.

HYPO's Dimensions have five possible types of ranges: binary, partially ordered sets of objects, ordered sets, intervals and number lines. For example, the *Security-Measures* range comprises sets of security measures partially ordered in terms of set inclusion. The RANGE slot of Figure 3 shows all of the possible measures. *Secrets-Disclosed-Outsiders,* Figure 4, has an interval range of from 0 to 10 000 000 outside disclosees. The comparison type specifies a method for comparing the focal slot values of two cases to which a Dimension applies. Different methods are appropriate for different types of ranges. For *Secrets-Disclosed-Outsiders,* HYPO simply tests which case involved the greater number of disclosees.

SHORT-TITLE: Secrets-Disclosed-Outsiders

CLAIMS: Trade Secrets Misappropriation

PREREQUISITES:
There is a corporate plaintiff
There is a corporate defendant
Plaintiff makes a product
Plaintiff and defendant compete
Plaintiff has product information
Plaintiff made some disclosures to outsiders

FOCAL SLOT PREREQUISITE:
Plaintiff made some disclosures to outsiders

FOCAL SLOT:
Plaintiff's Product Knowledge: NUMBER-DISCLOSEES

RANGE:
0 to 10 000 000

COMPARISON TYPE:
Greater-than versus Less-than

PRO PLAINTIFF DIRECTION: Less-than

FIGURE 4. The *Secrets-Disclosed-Outsiders*-Dimension.

A Dimension is a near miss if all of its prerequisites are satisfied, except the prerequisite associated with the focal slot. The focal slot prerequisite contains the crucial information that places a case somewhere along the Dimension's range. If the focal slot prerequisite is the only one missing, then the case is potentially close to cases indexed by the Dimension. Near misses indicate Dimensions that the user should consider because they index cases that could be cited in the argument if only small changes were made in the problem, changes that would place the problem somewhere along the Dimension's range. *Secrets-Disclosed-Outsiders,* shown in Figure 4, is a near miss with respect to *Amexxco* because, of all its prerequisites, only the focal slot prerequisite, Plaintiff-made-some-disclosures-to-outsiders, is missing.

6.3. CASE KNOWLEDGE BASE

All 30 cases in HYPO's CKB are represented in Legal Case Frames. At the time of entry into the CKB, each case was analysed by HYPO to determine the Dimensions that apply. The applicable Dimensions were then saved and listed in the top level case frame's DIMENSIONS-LIST. These Dimensions serve to index the case in the CKB. A general catalogue of all objects in HYPO, the Global Catalogue, has a list of pointers to all 30 cases' top level frames, a standard inverted file technique. When HYPO seeks all cases in the CKB that evidence a particular factor, for example, disclosures to outsiders, it runs through the Global Catalogue list extracting pointers to all cases with the *Secrets-Disclosed-Outsiders* Dimension in the top level case frame DIMENSIONS-LIST.

7. HYPO's processing steps

HYPO reasons by comparing problem situations with cases in its CKB and drawing inferences from the comparisons. The process is one of winnowing and clustering the cases in the CKB to make explicit their relationships in an argument about the problem. There are nine steps:

(1) Analyse the problem situation Dimensionally.
(2) Retrieve relevant cases from the CKB.
(3) Select relevant cases that are the most on point to the problem.
(4) Select most on point cases that are best for each side to cite.
(5) Compute distinctions between the best cases and the problem.
(6) Identify counter-examples to the best cases.
(7) Evaluate and summarize overall argument.
(8) Generate 3-Ply Arguments citing the best cases, distinctions and counter-examples.
(9) Generate hypotheticals to strengthen or weaken arguments.

HYPO's process culminates with argument outputs like those illustrated for the *Amexxco* problem. The following sections describe each step and illustrate it in the context of that problem.

7.1. DIMENSIONAL ANALYSIS AND RETRIEVAL OF RELEVANT CASES

In HYPO, relevant similarities are shared Dimensions. The set of relevant similarities, $S(c1, c2)$, between two cases, $c1$ and $c2$, is the intersection of the set of Dimensions that apply to $c1$ and the set of Dimensions that apply to $c2$.

Cases are said to be "on point" to a problem p, if and only if they are "relevantly similar" to the problem, i.e. $S(c1, c2)$ is not empty. The set of all cases c_i such that c_i is a member of the CKB and $S(p, c_i)$ is not empty is the set of cases that are on point to a problem p, $OP(p)$.

HYPO also employs a more relaxed definition of relevancy to identify cases that could be relevant to the analysis of the problem. The definition employs the concept of a near miss Dimension. The set of potentially relevant similarities between problem p and case $c2$, $PS(p, c2)$ is the intersection of the set of Dimensions that apply or are near misses to p and the set of Dimensions that apply to $c2$. The set of cases that are potentially on point to a problem p, $POP(p)$, is the set of all cases c_i such that c_i is a member of the CKB and $PS(p, c_i)$ is not empty.

Operationally, HYPO determines the sets $OP(p)$ and $POP(p)$ through Dimensional Analysis. First, HYPO tests each Factual Predicate on the problem's underlying Legal Case Frames and stores the results in the Interpretation Frame. Then, for each Dimension, HYPO compares the Dimension's prerequisites to the problem's Factual Predicate values to determine if the Dimension applies or is a near miss. Finally, HYPO retrieves any case in the CKB that is indexed by any Dimension that applies or is a near miss with respect to the problem.

The results of the Dimensional Analysis are stored in a Case Analysis Record (CAR). For the *Amexxco* problem, HYPO prepared the CAR shown in the Appendix. The INTERP-FRAME slot has a pointer to the completed Interpretation Frame for *Amexxco*. APPLICABLE-DIMENSIONS lists the Dimensions that apply to

Amexxco including *Security-Measures. Secrets-Disclosed-Outsiders* is listed in NEAR-MISS-DIMENSIONS. Pointers to the cases of $OP(p)$ and $POP(p)$ are contained the the CLAIM-LATTICES and EXTENDED-CLAIM-LATTICES slots, ordered, as described below, according to how on-point they are.

Perusing the CAR, one sees that HYPO has determined that three legal claims apply to the *Amexxco* case, trade secrets misappropriation, breach of nondisclosure/noncompetition agreement and breach of contract. That is, for each of these claims, there is at least one CKB case involving the claim which is indexed by at least one of *Amexxco's* five applicable Dimensions.

The following discussion is focused only on the trade secrets misappropriation claim and the cases and Dimensions regarding it. Since the only cases in the CKB to which the *Consideration* Dimension applies are contract cases, HYPO treats it as a contracts related Dimension. That Dimension is not involved in HYPO's treatment of the trade secrets claim in *Amexxco*.

7.2. SELECTION OF MOST ON POINT CASES

Saying that one case, c_i, is more on point than another case, c_k, means that the set of relevant similarities between the problem p and c_k, $S(p, c_k)$ is a proper subset of $S(p, c_i)$.

Cases that are most on point to the problem are the most relevantly similar of all the on-point cases. The set of most on point cases, *MOP*, is defined as the set of all cases, c_i such that for each c_i, there is no on-point case more on point than c_i. In other words, *MOP* is the set of all cases, c_i, such that c_i is on point and for all other on-point cases c_k, either c_i and c_k have no similarities in common with the problem situation, c_i is as or more on point than c_k, or neither is more on point than the other.

The set of potentially most on point cases, *PMOP*, is the set of all cases, c_i, such that for each c_i, there is no other case c_k in *POP* for which $PS(p, c_i)$ is a proper subset of $PS(p, c_k)$.

Operationally, HYPO sorts relevant and potentially relevant cases using Claim Lattices and extended Claim Lattices. A Claim Lattice is a directed acyclic graph. Every node contains pointers to parent and children nodes, a list of Dimensions and a list of cases. The maximal root node's list of Dimensions (i.e. the root's Dimension list) contains all of the Dimensions that apply to the problem. Each node contains all cases in the CKB such that the intersections of the root's Dimension list and the lists of Dimensions applicable to each case in the node are the same. The root node's case list contains all cases in the CKB that have the same set of applicable Dimensions as the problem. In effect, each node defines an equivalence class of cases that are equally on point with respect to the problem. An extended Claim Lattice is like a regular Claim Lattice except that its root's Dimension list also contains Dimensions that are near misses for the problem. In both kinds of Claim Lattices, the Dimension list of each node contains only Dimensions that are applicable to all of the cases in that node. For the *Amexxco* problem, HYPO generated the Claim Lattice shown in Figure 5. The root node represents the four trade secrets Dimensions that apply to the problem. It and successor nodes contain pointers to 13 cases that are on point. Each node shows the Dimensions that the included cases share with the problem and the winners, plaintiff (π) or defendant

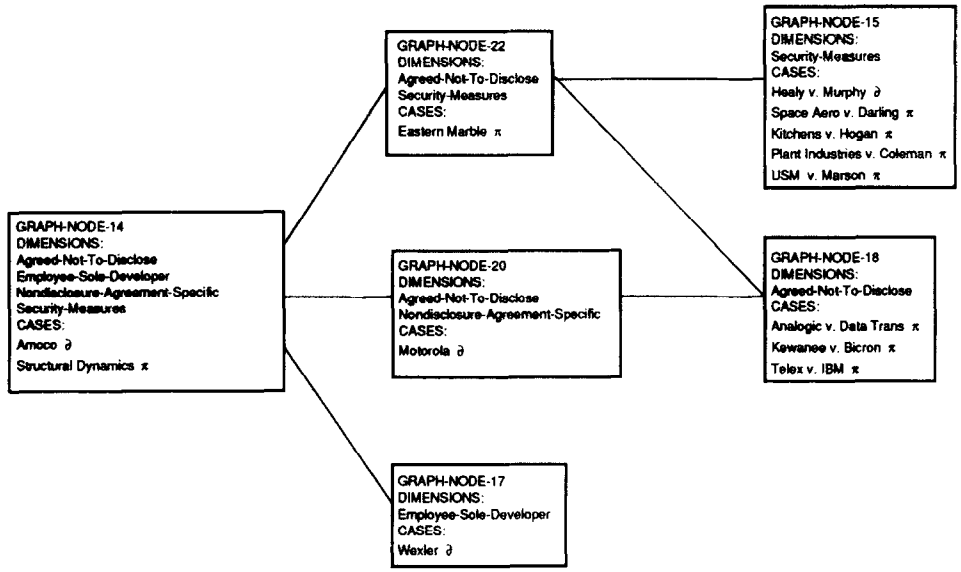

FIGURE 5. Claim Lattice for the *Amexxco* problem: the root node represents the *Amexxco* current fact situation and the four Dimensions that apply to it. The root node and successor nodes contain 13 cases that are on point. Each node shows the Dimensions that the included cases share with the problem and the winners, plaintiff (π) or defendant (δ). Cases in nodes closer to the root are more on point. Leaf nodes contain least on point cases. The *Amoco* case in the root node is defendant's most on point case. Plaintiff's most on point case is *Structural Dynamics*.

(δ). Cases can be inserted into the Claim Lattice in any order. Inserting a case into the Claim Lattice involves comparing the case with the nodes in a depth first search. The comparison checks the extent to which the set of Dimensions shared by the case and the cfs "covers" the node's list of Dimensions, that is, whether in a set comparison sense, it is greater than, equal to, less than, or an overlap of the node's list of Dimensions. The case is inserted just above any nodes containing cases whose set of shared Dimensions is a proper subset of the case's. The algorithm is described in detail in Ashley (1990), Chapter 8.

Having ordered the relevant cases in a Claim Lattice, selecting the most on point cases is simple. The closer a node is to the root, the more on point its cases are. Starting with the root, HYPO traverses each branch of the Claim Lattice until it finds a case or cases won by a side. Those cases are most on point for that side. In Figure 5, HYPO does not have to go far to find the plaintiff's and defendant's most on point cases; they are in the root node. *Structural Dynamics* is pro-plaintiff and more on point than any other pro-plaintiff case. Pro-defendant *Amoco* is more on point than any other pro-defendant case.

Comparing how on point cases are relative to a problem, does not entail comparing the number of Dimensions they share with the problem, but the overlaps of the sets of Dimensions they share with the problem. In Figure 5, neither *Motorola* nor *Wexler* is more on point than the other even though the former shares two Dimensions with the problem and the latter only one. The sets of Dimensions each shares with the problem are different. *Motorola* is more on point than all of its successor cases in the Claim Lattice, however.

HYPO stores the results of the selection process in the Case Analysis Record. Most on point cases are sorted by claim and contained in the MOST-ON-POINT-CASES-BY-CLAIM slot. The potentially most on point cases, garnered from the extended Claim Lattice (not shown) are stored in the MOST-ON-POINT-NEAR-MISS-CASES-BY-CLAIM slot. Note that *Structural Dynamics* is also a pro-plaintiff potentially most on point case.

7.3. SELECTING BEST CASES TO CITE

A side in an argument should lead off with its best cases. Since the most on point cases comprise the cases in the CKB that share with the problem the greatest overlap of factors, they are candidates for the best cases to cite for each side in an argument how to decide the problem. There is one qualification, however. A best case should share with the problem at least one Dimension that favored the case's winner and the side for whom the case is cited. More specifically, the best cases for a side to cite are those of the side's most on point cases that have at least one applicable Dimension favoring the side. The qualification insures that HYPO will not make a point for a side by citing a case that has no factors which favor the side. Sometimes, a most on point case shares with the problem only factors that favor the opponent. Such a case makes a good counter-example (i.e. it shows that those pro-opponent factors do not necessitate the opponent's winning) but it would not be a good case to lead off an argument for a side. In the *Amexxco* problem, the best cases to cite happen to include both of the most on point cases (compare the MOST-ON-POINT-CASES-BY-CLAIM and BEST-CASES-BY-CLAIM slot values for Trade-Secrets-Misappropriation claims in the CAR.) In the *Amoco* case, for example, the plaintiff adopted security measures and had a nondisclosure agreement with the defendant, both of which factors favor the plaintiff. Similarly, *Structural Dynamics* share with the problem at least one Dimension that favors the defendant winner of that case.

7.4. COMPUTING DISTINCTIONS

The relevant differences between a problem and a case are reasons why they should have different results. Assume that the outcome of the cited case favored the plaintiff. In HYPO, the set of relevant differences between the problem and cited case is the union of three sets: (1) the pro-defendant Dimensions that apply only to the problem; (2) the pro-plaintiff Dimensions that apply only to the cited case; and (3) the shared Dimensions that favor the plaintiff more strongly in the cited case than in the problem.

HYPO has procedures for distinguishing a case from a problem. It applies the procedures to each sides' best cases (and also to any counter-examples as described below) and records the results in an Argument Record.

Argument Records record information about how a particular case can be used in an argument about a particular problem, including its similarities and differences relative to the problem, counter-examples and hypothetical variations that it motivates. HYPO generates an Argument Record for each best case to cite, for each case manually selected by the user for a 3-Ply-Argument and for certain counter-examples. HYPO generates all of the information contained in an Argu-

ment Record dynamically. The Argument Records for the *Amexxco* problem are listed in the POINTS slot of the CAR. The Argument Record for the *Eastern Marble* case, one of the cases selected manually by the user, is shown below. HYPO fills in the similarities between the *Amexxco* cfs and the cited case *Eastern Marble* in the SHARED-DIMENSIONS slot.

NAME: Arg-Rec-2

CFS: Amexxco Production Co. v. Gwhiz

SIDE-1: Plaintiff

CLAIM: Trade-Secrets-Misappropriation

CITED-CASE: Eastern Marble Products Corp. v. Roman Marble, Inc.

SHARED-DIMENSIONS:
(*Security-Measures*
Agreed-Not-To-Disclose)

SIDE-2-DISTINCTIONS:
((Weaker Amexxco Production Co. v. Gwhiz
Eastern Marble Products Corp. v. Roman Marble, Inc.
Security-Measures
Less (Minimal-Measures) (Minimal-Measures
Employee-Nondisclosure-Agreements
Restrictions-On-Entry-By-Visitors))
(Strong Amexxco Production Co. v. Gwhiz
Employee-Sole-Developer))

MOP-COUNTER-EXMPL: (Amoco Production Co. v. Lindley)

AS-OP-COUNTER-EXMPL: Nil

POTENT-COUNTER-EXMPL: Nil

BOUNDARY-COUNTER-EXMPL: Nil

SIDE-1-HYPOS: Nil

SIDE-2-HYPOS: Nil

RESPONSIVE-ARG-RECS: (Arg-Rec-1)

HYPO records the distinctions between a case and a problem in the corresponding Argument Record's SIDE-2-DISTINCTIONS slot. The first distinction list in that slot of Argument Record 2 indicates that *Amexxco* is weaker than *Eastern Marble* along the *Security-Measures* Dimension because Amexxco's security measures (i.e. minimal measures) were less than Eastern Marble's (i.e. minimal measures plus employee nondisclosure agreements and restrictions on visitor entry.) HYPO compares cases along shared Dimensions such as *Security-Measures* using the focal slot, range, comparison type and pro-plaintiff direction information contained in the Dimension. The second distinction in Argument Record 2 indicates that a Dimension favoring the defendant, *Employee-Sole-Developer,* applied in *Amexxco* that does not apply in *Eastern Marble.* HYPO finds such differences as these by comparing the cases' top-level Legal Case Frame DIMENSIONS-LISTS and determining whether the unshared Dimensions favor the opponent of the side citing the case.

7.5. IDENTIFYING COUNTEREXAMPLES

HYPO identifies four kinds of counter-examples. If a side cites a precedent as a case example justifying a decision in its favor, a counter-example is a precedent that is contrary to the cited precedent (i.e. the opposite side won the counter-example) and that, in some well-defined sense, parries the thrust of the justification. There are four senses in which a counter-example parries a precedential justification, each corresponding to one of the four types of counter-examples:

(1) Trumping
(2) Potentially trumping
(3) As-on-point or partial
(4) Boundary.

A trumping counter-example shares all of the relevant similarities of the precedent to the problem and some additional ones, as well. This kind of counter-example "trumps" the cited precedent. It says, in effect, that when the extra similarities are taken into account, the precedent should not be followed; the opposite outcome should result. More precisely, the set of trumping counter-examples, $TCX(s, cc)$, that can be cited by a side s in response to a case cc, cited by the opponent, is defined as the set of all most on point cases, c_i, such that the outcome of c_i favors s and c_i is more on point than cc. This definition corresponds to the intuitive idea of a trumping counter-example conveyed in Figure 1.

A potential trumping counter-example is a case that would trump a cited precedent in a somewhat different problem situation. The set of potentially trumping counterexamples, $PTCX(s, cc)$, that can be cited by a side s in response to a case cc, cited by the opponent, is defined as the set of all potentially most on point cases, c_i, such that the outcome of c_i favors s and c_i is more on point than cc relative to a hypothetically modified problem p'. Hypothetical problem p' is equivalent to p plus the near misses; each Dimension d that is a near miss with respect to p is applicable to p' (i.e. p' corresponds to the root node of the extended Claim Lattice.) Each of these potential counter-examples motivates a suggested hypothetical modification of the problem in which a side would have a new counter-example with which to trump the opponent's point.

As-on-point or partial counter-examples share the same set as, or some non-empty subset of, the set of Dimensions that the cited precedent shares with the problem. If the set is the same, the counter-example is as on point as the cited case and creates doubts about the cited case's reliability. A partial counter-example shows, somewhat more weakly than a trumping or as-on-point counter-example, that the shared similarities do not always lead to the same outcome as the cited case.

A boundary counter-example shows that a particular similarity is not a significant reason for deciding a case. It is a case in which the same similarity would appear to favor a side even more than in the cited case but where the opponent won nevertheless. Concisely, the set, $BCX(s, cc)$ of all cases that a side s can cite as boundary counter-examples to a cited case cc is the set of all on-point cases, c_j, such that s won c_j and there is some Dimension, d, such that c_j and cc share d with the problem, and the magnitude of d in c_j is greater in favor of s's opponent than in cc. In other words, although d is a relevant similarity between cc and p, even greater magnitudes of d in a case do not require coming to the same decision as in cc; witness the boundary counter-example, c_j.

Operationally, HYPO looks for counter-examples to each side's best cases to cite and to any case selected manually by the user for a 3-Ply-Argument. To identify trumping and as-on-point counter-examples, HYPO compares the cited case with each of the other side's most on point cases. For instance, HYPO compared plaintiff's best case *Structural Dynamics* and the manually selected *Eastern Marble* with the defendant's most on point case *Amoco*. It compared each plaintiff's case's set of similarities to the problem with that of *Amoco*. Since *Amoco* and *Structural Dynamics* had equivalent sets of similarities, *Amoco* is an as-on-point counter-example. Since *Eastern Marble's* set of similarities was a proper subset of *Amoco's*, *Amoco* is a trumping counter-example to *Eastern Marble*. HYPO also found that *Structural Dynamics* trumped defendant's manually selected cases, *Motorola* and *Wexler*. HYPO employs similar procedures for identifying potentially trumping counter-examples. It compares the cited case with the opponent's potentially more-on-point cases (in the MOST-ON-POINT-NEAR-MISS-CASES-BY-CLAIM slot of the Case Analysis Record.) It found that the *Structural Dynamics* case was potentially more-on-point than *Amoco* because, although they are equally on point, *Structural Dynamics* has one additional Dimension, *Brought-Tools*, which is a near-miss in the *Amexxco* problem.

HYPO records all counter-examples to a cited case in the Argument Record for that case. Argument Record 2 indicates that HYPO has found a trumping counter-example, the *Amoco* case, to use against the *Eastern Marble* case. (Note the MOP-COUNTER-EXMPL slot value.)

HYPO creates argument links between an Argument Record for a cited case and an Argument Record for a trumping, as-on-point or boundary counter-example. (If the counter-example does not already have its own Argument Record, HYPO creates one.) Later, in generating 3-Ply Arguments, HYPO will follow these links in generating responses to points. For example, HYPO generated an Argument Record (Arg-Rec-1—shown below) for the *Amoco* case. In filling out the Argument Record for the *Eastern Marble* case, it draws a link to Arg-Rec-1 to indicate that *Amoco* is a counter-example. The RESPONSIVE-ARG-RECS slot of the *Eastern Marble* Argument Record (Arg-Rec-2) is filled in with a pointer to Arg-Rec-1 to indicate that a response to citing *Eastern Marble* is citing its counter-example, *Amoco*. The Argument Record for the counter-example (Arg-Rec-1) shows a comparison between the *Amoco* case and the *Amexxco* problem. The two are very close; there are no distinctions. Similarly, Arg-Rec-1's RESPONSIVE-ARG-RECS slot has a pointer to the Argument Record for *Structural Dynamics* (Arg-Rec-0, not shown) to indicate that one way to respond to citing *Amoco* is by citing *Structural Dynamics* as an as-on-point counter-example. (Note the AS-OP-COUNTER-EXMPL slot.) The POTENT-COUNTER-EXMPL slot records the fact that *Structural Dynamics* is potentially more on point than *Amoco*.

NAME: Arg-Rec-1

CFS: Amexxco Production Co. v. Gwhiz

SIDE-1: Defendant

CLAIM: Trade-Secrets-Misappropriation

CITED-CASE: Amoco Production Co. v. Lindley

SHARED-DIMENSIONS:
(*Secuity-Measures*
Nondisclose-Agreement-Specific
Employee-Sole-Developer
Agreed-Not-To-Disclose)

SIDE-2-DISTINCTIONS: Nil

MOP-COUNTER-EXMPL: Nil

AS-OP-COUNTER-EXMPL: Structural Dynamics Research Corp. v. Engineering Mechanics Research Corp.

POTENT-COUNTER-EXMPL: Structural Dynamics Research Corp. v. Engineering Mechanics Research Corp.

BOUNDARY-COUNTER-EXMPL: Nil

SIDE-1-HYPOS: Nil

SIDE-2-HYPOS: Structural Dynamics Research Corp. v. Engineering Mechanics Research Corp.

RESPONSIVE-ARG-RECS: Arg-Rec-0

Boundary counter-examples are a way to respond to an asserted similarity, to show that the similarity does not necessarily mean that an opponent should win. The procedure for finding boundary counter-examples is to check if the Dimensions shared by the cfs and cited case (from the Argument Record SHARED-DIMENSIONS slot) index any cases won by Side-2 even though they are weaker for Side-2 than the cited case. Boundary counter-examples are sought only for Dimensions whose ranges are number lines, ordered sets of more than two members, or partially ordered sets. These range types support a sense in which a case can be an extreme example of the magnitude of a Dimension. There happen to be no boundary counter-examples in connection with the *Amexxco* case. There would have been a boundary counter-example if the CKB contained a case that had all or more of the security measures of *Eastern Marble* but the defendant had won. HYPO would cite such a case in response to *Eastern Marble* to show that security measures, though favoring plaintiff, do not guarantee a pro-plaintiff result.

7.6. EVALUATING AND SUMMARIZING OVERALL ARGUMENTS

HYPO evaluates the opposing overall arguments by determining whether either side alone can cite untrumped best cases. These are cases with respect to which there are no better cases for the opponent to cite as trumping counter-examples. More concisely, the set of untrumped best cases for a side s to cite, *UBCC*, is the set of cases, c_i, such that c_i is a most on point case, s won c_i, c_i shares some Dimension with the problem that favors s and there is no trumping counter-example for s's opponent to cite. If both sides have untrumped best cases to cite, HYPO does not select a winner but treats both sides as having strong arguments.

HYPO reports its evaluation in a Citation Summary that describes which side has the stronger position and lists each side's untrumped case. For the *Amexxco* problem, HYPO treats both sides as having strong arguments because both have untrumped points to make.

7.7. GENERATING 3-PLY ARGUMENTS

HYPO outputs a 3-Ply Argument for each side's best cases to cite and for any cases selected by the user. The Argument Records are blueprints for 3-Ply Arguments. For example, Argument [2] was generated from Arg-Rec-1 and Argument [3] came from Arg-Rec-2. HYPO makes the first ply, Side-1's point, by citing the Argument Record's cited case and drawing an analogy between it and the cfs. As a case cited in a point, the salient features are its relevant similarities, the Dimensions shared by the case and problem. These relevant similarities are stored in the Argument Record's SHARED-DIMENSIONS slot. HYPO draws the analogy to the cfs by stating a factual description associated with each of the stored Dimensions.

The second ply, Side-2's response, consists of distinguishing the case cited in the point and citing counter-examples to it. The salient features of a case as distinguished from the cfs are its relevant differences as defined above, certain unshared Dimensions and differences in magnitude of shared Dimensions. HYPO recites Side-2's distinctions from the Argument Record's SIDE-2-DISTINCTIONS slot.

In the response, HYPO also cites each trumping, as-on-point and boundary counter-example that it previously stored in the Argument Record. A trumping counter-example's salient features are the extra Dimensions that the counter-example shares with the problem that the cited case does not. In describing the trumping counter-example, HYPO emphasizes the counter-example's extra shared Dimensions relative to the cited case, computed by taking the set difference of the SHARED-DIMENSIONS in the Argument Record for the cited case and that for the counter-example. For instance, in describing *Amoco* in Argument [3] as a counter-example to trump *Eastern Marble,* HYPO emphasizes *Amoco*'s extra shared Dimension: the employee defendant was the sole developer of plaintiff's product. For a boundary counter-example, the salient feature is the Dimension whose magnitude is more extreme in the counter-example than in the cited case.

The third ply, Side-1's rebuttal, entails distinguishing any of the counter-examples from the cfs. The Argument Record for the counter-example, which HYPO linked to the Argument Record for the cited case, contains those distinctions in its SIDE-2-DISTINCTIONS slot. In Argument [2]'s rebuttal, the distinctions between the *Structural Dynamics* counter-example and *Amexxco* come from the Argument Record for *Structural Dynamics* (Arg-Rec-0, not shown but referred to in Arg-Rec-1 as a responsive Argument Record.)

In order to generate the case descriptions in the 3-Ply Arguments, HYPO takes the list of Dimensions that represent the salient features of the case in a particular argument context. Each Dimension has associated canned textual phrases for asserting that the Dimension applies to a case or that its magnitude is more or less than in another case. HYPO concatenates statements corresponding to the listed Dimensions. This method provides some measure of flexibility in describing cases even with canned textual phrases.

7.8. SUGGESTING AND POSING HYPOTHETICALS

Finally, HYPO suggests hypothetical modifications of the problem situation in which a side's argument would be strengthened or weakened. The modifications involve adding facts to the cfs so that Dimensions, formerly near misses with respect to the

problem, become applicable. As a result of the modifications, new cases become relevant opening up new possible points and responses. Such cases are the targets of HYPO's suggested modifications; modifying the cfs as suggested makes it more like a target case.

There are two sources of target cases for the modifications. First, the most on point near miss cases in the Case Analysis Record are targets for hypothetical modifications that lead to new argument points. Second, the potentially more-on-point counter-examples listed in an Argument Record are targets leading to new responses.

As examples of the first source, each of the suggested hypotheticals in HYPO's output for *Amexxco* in Section 5, corresponds to a most on point near-miss trade secrets case from the *Amexxco* Case Analysis Record. Some targets would strengthen the plaintiff's position; others, like the *Crown Industries* case favor the defendant.

A second source of targets for hypotheticals leads to new responses. HYPO suggests posing a variation of the cfs in which the counter-example would trump the cited case. Operationally, if the Argument Record indicates that there are any potentially more-on-point counter-examples to a cited case, HYPO appends a note to the 3-Ply Argument offering to pose a hypothetical variant of the cfs in which Side-2's response would be bolstered. Argument [2], for example, contains such a note. It says that the plaintiff's response would be strengthened if its former employee had brought product development information, and it cites the *Structural Dynamics* case which, as we have seen, is a potentially more-on-point counter-example to *Amoco*.

The salient features of a case cited as a target of a hypothetical variant are some of the target case's Dimensions that are near misses with respect to the problem. HYPO generates the description of the target case from two lists of Dimensions, those that apply to the target (from the target's DIMENSIONS-LIST) and those that are near misses in the cfs (from the CAR NEAR-MISS-DIMENSIONS slot.) HYPO takes the set intersection of those lists of Dimensions, screens them to select only Dimensions that favor the side for whom the hypothetical is suggested and uses the result to describe the salient features of the target. In the Argument [2] note, for example, HYPO emphasizes the *Brought-Tools* Dimension as the salient feature of the *Structural Dynamics* case in this context.

When the user selects a suggested hypothetical, HYPO has methods for generating a modified version of the cfs. HYPO creates a specification, a list of changes such as the Dimensions that should be added to the cfs or along which a case should be strengthened or weakened and which side the change should favor. Given a specification, HYPO makes a copy of the case to be modified. The copy is identical except that the names of the Legal Case Frame instances are modified to avoid obliterating the case to be modified. Then HYPO implements the changes in the specification using five heuristics and their associated modification methods.

As described more fully in Rissland and Ashley (1986) and Ashley (1990), HYPO's five heuristics for modifying cases hypothetically are:

H1: Make a near miss Dimension apply;

H2: Strengthen or weaken a case along an applicable Dimension;

H3: Move a case along a related Dimension;

H4: Make a case extreme along a Dimension;

H5: Make a case into a near-win given a target.

Each heuristic employs modification methods. Some modification methods take a list of Factual Predicates corresponding to the missing prerequisites of a near miss Dimension and add corresponding Legal Case Frames or slot values to the case so that the Dimension applies. Other modification methods are like methods for comparing cases along a Dimension's range, except that instead of comparing focal slot values, they change the value. These methods are keyed to Dimensions' range and comparison types.

On the user's command, HYPO employs the heuristics to modify the cfs. For example, suppose the user takes up HYPO's suggestion in Argument [2]'s note that the plaintiff's response could be strengthened if the employee had brought product development tools to the defendant. Using the *Structural Dynamics* case as a target and using H1, HYPO can modify the cfs so that plaintiff can cite *Structural Dynamics* as a trumping counter-example. Pursuant to H1, one or more of the target's Dimensions that were near misses in the cfs, are made to apply in the modified cfs. In particular, the *Brought-Tools* Dimension is made to apply to the *Amexxco* problem. HYPO generates the new arguments and helps the user compare them with the previous arguments.

The result is a dramatic change in the balance of the argument. Now when HYPO generates a 3-Ply-Argument citing *Amoco* for the defendant, there is a new response:

> ⇐ Response for Plaintiff as Side-2:
> Counter-examples:
> Structural Dynamics Research Corp. v. Engineering Mechanics Research Corp. 401 F. Supp. 1102 (E.D. Mich. 1975), is more on point and held for plaintiff where it was also the case that: Plaintiff's former employee brought product development information to defendant.

Compare this response with the former response in Argument [2]. Now the plaintiff can trump the defendant's point. (Note also that in this different context, HYPO's description of *Structural Dynamics* changes to emphasize its newly salient aspect.) The new trumping counter-example tips the strength of the sides' arguments in favor of plaintiff. HYPO's new case summary says:

> On a claim for Trade Secrets Misappropriation, plaintiff can make a stronger argument.
> Plaintiff can cite the following cases for which there are no more on point counter-examples:
> Structural Dynamics Research Corp. v. Engineering Mechanics Research Corp. 401 F. Supp. 1102 (E.D. Mich. 1975)

The new situation is reflected in a new Claim Lattice for the modified version of the *Amexxco* problem in Figure 6. The root node now shows that the *Brought-Tools* Dimension applies to the problem. *Structural Dynamics* still has all of the Dimensions that apply to the problem. *Amoco*, however, has been demoted to a successor node. It is now less on point than *Structural Dynamics*. The *Analogic* and *Space Aero* cases have also become more on point than their former neighbors in Figure 5.

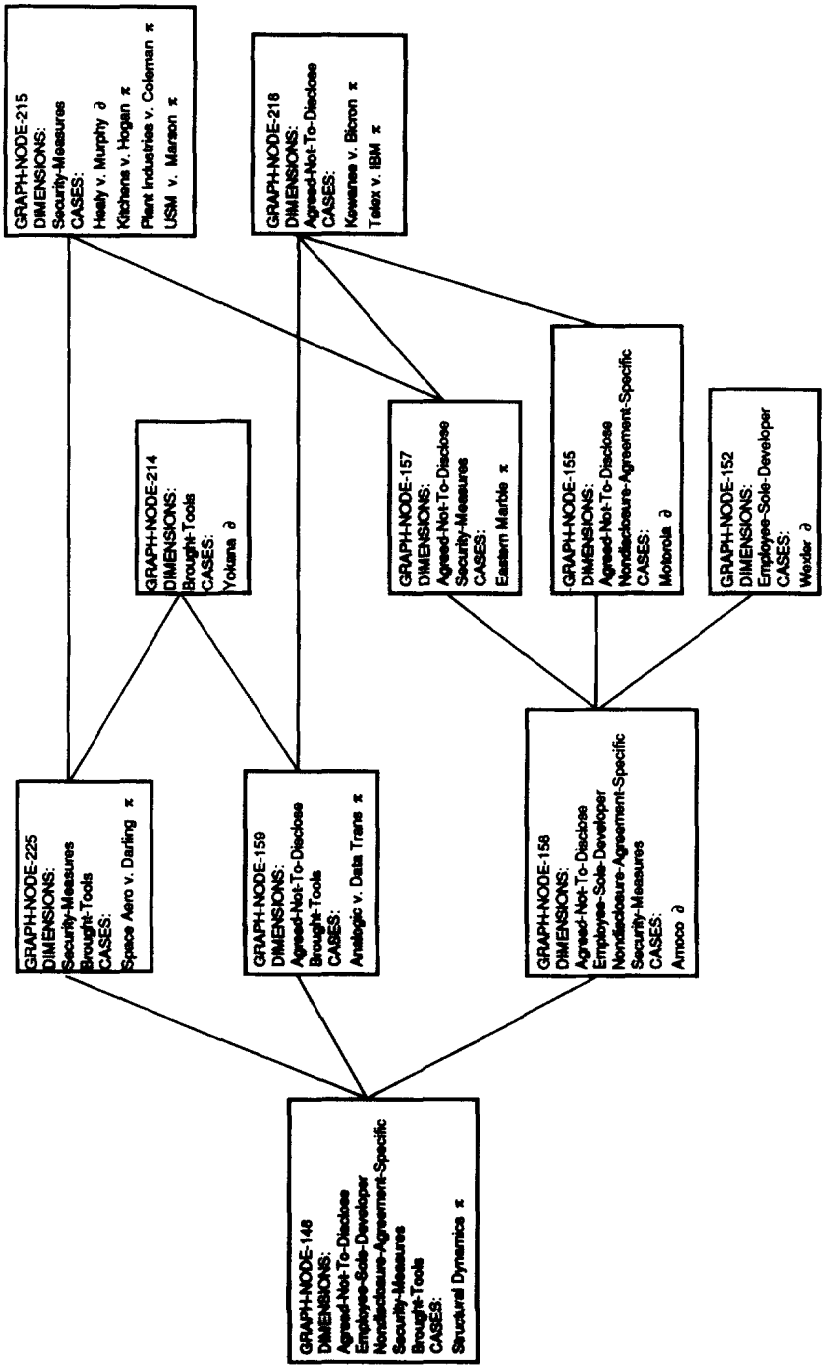

FIGURE 6. Claim Lattice for the hypothetically modified *Amexxco* fact situation: since the *Bought-Tools* Dimension now applies to the problem, *Structural Dynamics* trumps *Amoco*.

On the user's command, HYPO can also strengthen the defendant's position. For example, suppose the user takes up HYPO's suggestion that the defendant's position could be strengthened if the plaintiff had made disclosures to outsiders and in negotiations with the defendant. With heuristic H1, HYPO can modify the cfs so that the defendant can cite the *Crown Industries* case in its favor in a new point to the effect that the plaintiff should lose when it makes such disclosures. Taking *Crown Industries* as a target, generating the hypothetical involves making the cfs near miss Dimensions, *Secrets-Disclosed-Outsiders* and *Disclosure-in-Negotiations*, apply to the new cfs. In other words, the hypothetical supposes that Amexxco disclosed its secrets to seven outsiders and in negotiations with the defendant as in *Crown Industries*.

One hypothetical modification leads to others. Adding new points for one side opens up new potential responses for the other. The user can follow a series of hypothetical modifications as far as there are targets. Suppose that the *Amexxco* cfs were modified, as suggested above, to include disclosures by Amexxco to outsiders so that the *Crown Industries* case could be cited in a point for the defendant. HYPO would suggest a new way for the plaintiff to make a response. The target is a new case that would then become a potentially trumping counter-example to *Crown*: the *Data General* case where the plaintiff won where all of the disclosees agreed to maintain confidentiality. In *Data General*, an extra Dimension *Outsider-Disclosures-Restricted*, applies. Once disclosures are added to *Amexxco*, *Outsider-Disclosures-Restricted* becomes a near miss. Using H1, the *Amexxco* problem can be modified yet again to make *Outsider-Disclosures-Restricted* apply. Plaintiff can then trump the *Crown Industries* case by citing *Data General* in response.

HYPO can also modify the cfs to make it stronger or weaker along an applicable Dimension. For example, assume that the new cfs is the modified *Amexxco* problem with two newly applicable Dimensions: there were disclosures to seven outsiders (i.e. *Secrets-Disclosed-Outsiders* applies) and all of the disclosees agreed to maintain confidentiality (i.e. *Outsider-Disclosures-Restricted* applies.) Using H2, Hypo can make the newly modified cfs weaker for the plaintiff along the *Secrets-Disclosed-Outsiders* Dimension by increasing the number of disclosures to 100 outsiders (as in another target case from the CKB). Using H4, the cfs could be further weakened to an extreme: 10 000 000 disclosures to outsiders, the maximum value of the Dimension's range (not an unrealistic figure in these days of mass marketing of software). Implicitly, the hypothetical modification raises the question, "Is a secret still confidential if 10 000 000 disclosees know it, even if they have agreed to maintain confidentiality?" Although HYPO has no understanding of what "secret" or "confidential" mean, it does know that *Secrets-Disclosed-Outsiders* and *Outsider-Disclosures-Restricted* are related Dimensions that favor opposite sides and that moving the cfs to extremes along one of the related Dimension's (using heuristic H3) is likely to create an interesting hypothetical that questions the significance of the other Dimension (Ashley, 1990, Ch. 8).

8. Discussion

HYPO evaluates a problem by comparing and contrasting it with cases in its CKB. The 3-Ply Arguments and suggested hypotheticals make these comparisons explicit.

They present two competing adversarial views of the problem and hypothetical suggestions for strengthening one side or the other.

The hypotheticals focus a user on additional factors to look for in a problem situation and the accompanying arguments explain why the new factors are significant. They also allow a user to explore the consequences on an argument of failing to prove that a factor is present. Finally, hypotheticals varying the magnitudes of factors perhaps to extremes, can be used to argue that a factor is or is not very important.

HYPO's computational definitions of relevance and salience make it possible to compare and contrast cases and hypotheticals. This discussion has presented and illustrated computational definitions of some important concepts involving relevance: relevant similarities and differences, more on point, most on point cases, best cases to cite, four kinds of counter-examples, and targets for hypotheticals. There are also computational definitions for describing the salient features of a case selected for any of the four roles it can play in HYPO's arguments, as a case: (1) cited in a point; (2) distinguished in a response; (3) cited as a counter-example; or (4) suggested as a target for a hypothetical. These definitions allow HYPO to select and describe cases in a contextually sensitive way.

The context of the argument in which the cases are to be used comprises the particular problem facts, the side that is arguing, the other cases in the CKB and their facts, and the role of the case in the argument.

HYPO's computational definitions for selecting relevant cases take all of these contextual elements into account. HYPO judges a case's relevance by the opportunities it presents for making argument points and responses: whether the case could be used to trump a point, or show that a factor is not always important, or show that a small change in facts could lead to a new point or a new response. The program computes those opportunities dynamically, for any particular problem and body of cases in the CKB. Its process projects the CKB onto the cfs's particular facts and groups cases into classes of equivalent on pointness. The quality of a match is assessed not only by how close a case is to the problem facts but how close other cases are. If other comparably on point cases compete, counter-example relations are made explicit. In selecting cases, HYPO views the problem from both sides. It determines the best use a side can make of a case, whether a side should lead off with the case in a point or reserve the case for use as a responsive counter-example. Selecting cases from the other side's viewpoint, HYPO trumps points and attacks similarities with counter-examples and suggests target cases for changing the analysis hypothetically.

Having selected a case to cite in an argument role, HYPO also takes context into account in determining the aspects of the case that are salient. In each of the four roles, HYPO generates a different salient description of the case. HYPO's description of a single case varies with the case's argument role, the side for whom it is cited, and the particular problem facts.

In making these relevance and salience assessments, HYPO does not rely on the existence of a strong domain theory. By taking the argument context into account, it can evaluate some kinds of partial matches even though there is no causal theory for assessing differences. HYPO also avoids *a priori* hierarchies or numerical weighting schemes to represent a factor's import. The differences among cases have not been reduced to a number. HYPO's arguments play out the differences among cases

symbolically. Their significance lies in the argument-making opportunities they present.

HYPO's arguments compare favorably with those of actual judges and attorneys recorded in case opinions and briefs. Current fact situations based on four real legal cases were submitted to HYPO, and its 3-Ply Arguments and hypotheticals were compared with the points and responses made in the attorney's briefs and judge's opinions. They were compared in terms of whether HYPO's evaluation of the relative overall strength of parties' arguments tallied with the winner, whether HYPO made points and responses for each side citing the same prededents and in the same way as the judges and attorneys, and whether HYPO's citations and suggested hypotheticals raised the same issues as in the briefs and opinions. As reported in Ashley (1990) Chapter 10, there were some striking similarities between HYPO's outputs and the ways that the judges and attorneys cited and responded to cases.

There are other important kinds of legal arguments, other possible ways of justifying a resolution of competing factors, with which HYPO does not deal. An attorney often presents a more technical rationale or explanation of why a problem's competing factors should be resolved in favor of his or her client. These rationales may invoke statutory and other legal rules and predicates. They may also involve arguing by analogy to past cases in a different sense: mapping a rationale from a past case to the problem. Since HYPO was designed, work has progressed on modeling this more technical legal analysis. The program GREBE maps onto a problem the structure of a court's rationale for deciding a precedent (Branting, 1991). CABARET, a program that applies legal rules to the analysis of a problem, invokes a HYPO-like case-based reasoner with Dimensions to justify conclusions that the rules' legal predicates apply to a problem or not (Rissland & Skalak, 1991). Both efforts employ cases to evaluate problems but use generalizations differently to generate more abstract, technical explanations of the significance of similarities and differences among cases, with or without structural mapping to the cases. As programs apply more technical legal analysis, they must apply techniques for assessing relevant cases and significant facts in contextually sensitive ways and deal with the fact that there are competing explanations of almost any case. Inevitably, the programs will become computationally more complex (especially those involving structural mapping).

Just as the drafters of the Restatement relied on factors, so will AI find Dimensions useful tools where more abstract causal or logical definitions fail or become too computationally expensive to apply. HYPO's tractable methods for selecting cases and counter-examples will assist programs to select those cases in which to invest computational resources to apply more analytical methods. HYPO's approach to computing a case's salient features differently depending on the argument context will help programs decide which similarities or differences are important enough to expend computational resources to explain or rebut.

9. Conclusion

HYPO is a case-based reasoning system that evaluates problems by comparing and contrasting them with cases from its Case Knowledge Base. It generates legal arguments citing the past cases as justifications for legal conclusions about who

should win in problem disputes involving trade secret law. HYPO's arguments present competing adversarial views of the problem and it poses hypotheticals to alter the balance of the evaluation. HYPO uses Dimensions as a generalization scheme for accessing and evaluating cases. HYPO's reasoning process and various computational definitions have been described and illustrated, including its definitions for computing relevant similarities and differences, most on point and best cases to cite, four kinds of counter-examples, targets for hypotheticals and the aspects of a case that are salient in various argument roles. These definitions enable HYPO to make contextually sensitive assessments of relevance and salience without relying on either a strong domain theory or *a priori* weighting schemes.

References

ASHLEY, K. D. (1988). Arguing by analogy in law: a case-based model. In D. H. HELMAN, Ed. *Analogy Reasoning: Perspectives of Artificial Intelligence, Cognitive Science, and Philosophy*. Dordrecht, The Netherlands. Kluwer.

ASHLEY, K. D. (1989). Defining salience in case-based arguments. In *Proceedings IJCAI-89*, pp. 537–542. Detroit: International Joint Conferences on Artificial Intelligence,

ASHLEY, K. D. (1990). *Modeling Legal Argument: Reasoning with Cases and Hypotheticals*. Cambridge, MA: MIT Press. (Based on Ph.D. thesis, University of Massachusetts, Computer and Information Science Department, 1987).

ASHLEY, K. D. & RISSLAND, E. L. (1987). Compare and contrast, a test of expertise. In *Proceedings AAAI-87*, pp. 273–278. Seattle: American Association for Artificial Intelligence.

ASHLEY, K. D. & RISSLAND, E. L. (1988*a*). A case-based approach to modeling legal expertise. *IEEE Expert*, **3**, 70–77.

ASHLEY, K. D. & RISSLAND, E. L. (1988*b*). Waiting on weighting: a symbolic least commitment approach. In *Proceedings AAAI-88*, pp. 239–244. St. Paul, MN: American Association for Artificial Intelligence.

BAREISS, E. R. (1989). *Exemplar-Based Knowledge Acquisition—A Unified Approach to Concept Representation, Classification, and Learning*. San Diego, CA: Academic Press.

BAREISS, E. R., PORTER, B. W. & WIER, C. C. (1987). PROTOS: an exemplar-based learning apprentice. In *Proceedings Fourth International Workshop on Machine Learning*, University of California at Irvine, pp. 12–23.

BRANTING, L. K. (1991). Building explanations from rules and structured cases. *International Journal of Man–Machine Studies*, **34**, 797–837.

CARBONELL, J. & VELOSO, M. (1988). Integrating derivational analogy into a general problem solving architecture. In *Proceedings of the Case-Base Reasoning Workshop*. Clearwater Beach, FL: Defense Research Projects Agency/Information Science and Technology Office.

CARBONELL, J. G. (1986). Derivational analogy: a theory of reconstructive problem solving and expertise acquisition. In R. S. MICHALSKI, J. G. CARBONELL & T. M. MITCHELL, Eds. *Machine Learning: An Artificial Intelligence Approach*, Vol. II. Los Altos, CA: Morgan Kaufmann.

GILBURNE, M. R. & JOHNSTON, R. L. (1982). Trade secret protection for software generally and in the mass market. *Computer/Law Journal*, **III**, 211–272.

HAMMOND, K. J. (1986). CHEF: a model of case-based planning. In *Proceedings AAAI-86*, pp. 267–271. Philadelphia: American Association for Artificial Intelligence.

HAMMOND, K. J. (1987). Explaining and repairing plans that fail. In *Proceedings IJCAI-87*. Milan, Italy: International Joint Conferences on Artificial Intelligence.

HAMMOND, K. J. (1989). *Case-Based Planning—Viewing Planning as a Memory Task*. San Diego, CA: Academic Press.

KOLODNER, J. L. (1983). Reconstructive memory: a computer model. *Cognitive Science*, **7**, 281–328.

KOLODNER, J. L., SIMPSON, R. L. & SYCARA-CYRANSKI, K. (1985). A process model of case-based reasoning in problem solving. In *Proceedings IJCAI-85,* pp. 284–290. Los Angeles: International Joint Conferences on Artificial Intelligence.

KOTON, P. (1988*a*). Reasoning about evidence in causal explanations. In *Proceedings AAAI-88,* pp. 256–261. St. Paul, MN: American Association for Artificial Intelligence.

KOTON, P. (1988*b*). *Using Experience in Learning and Problem Solving.* Ph.D. thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology.

MILGRIM, R. M. (1985). *Business Organizations, Milgrim on Trade Secrets,* vol. 12. New York: Matthew Bender.

NIMMER, R. T. (1985). *The Law of Computer Technology.* Boston: Warren, Gorham & Lamont.

RISSLAND, E. L. & ASHLEY, K. D. (1986). Hypotheticals as heuristic device. In *Proceedings AAAI-86,* pp. 289–297. Philadelphia: American Association for Artificial Intelligence.

RISSLAND, E. L. & SKALAK, D. B. (1989). Combining case-based and rule-based reasoning: A heuristic approach. In *Proceedings IJCAI-89,* pp. 524–530. Detroit: International Joint Conferences on Artificial Intelligence.

RISSLAND, E. L. & SKALAK, D. B. (1991). CABARET: statutory interpretation in a hybrid architecture. *International Journal of Man–Machine Studies,* **34,** 839–887.

STANFILL, C. & WALTZ, D. (1986). Toward memory-based reasoning. *Communications of the ACM,* **29,** 1213–1228.

STANFILL, C. W. (1987). Memory-based reasoning applied to English pronunciation. In *Proceedings AAAI-87,* pp. 577–581. Seattle: American Association for Artificial Intelligence.

Appendix: Case Analysis Record for *Amexxco*

CASE: Amexxco Production Co. v. Gwhiz

INTERP-FRAME: Interp-Frame:Amexxco Production Co. v. Gwhiz

APPLICABLE-DIMENSIONS:
(*Consideration*
Security-Measures
Nondisclose-Agreement-Specific
Employee-Sole-Developer
Agreed-Not-To-Disclose

NEAR-MISS-DIMENSIONS:

(*Brought-Tools*
Bribe-Employee
Secrets-Disclosed-Outsiders
Competitive-Advantage
Vertical-Knowledge
Disclosure-In-Negotiations)

CLAIM-LATTICES:
((Trade-Secrets-Misappropriation Graph-Node-14)
(Breach-Nondisclose-Noncompete-Agreement Graph-Node-7)
(Breach-Of-Contract Graph-Node-0))

EXTENDED-CLAIM-LATTICES:
((Trade-Secrets-Misappropriation Graph-Node-47)
(Breach-Nondisclose-Noncompete-Agreement Graph-Node-39)
(Breach-Of-Contract Graph-Node-32))

MOST-ON-POINT-CASES-BY-CLAIM:
((Breach-Of-Contract
(Plaintiff Laff v. John O. Butler Co.)
(Defendant
Dougherty v. Salt
Hancock Bank & Trust Co. v. Shell Oil Co.
Akirksey v. Kirksey))
(Breach-Nondisclose-Noncompete-Agreement
(Plaintiff
Structural Dynamics Research Corp. v. Engineering Mechanics Research Corp.)
(Defendant
Motorola, Inc. v. Fairchild Camera and Instrument Corp.))
(Trade-Secrets-Misappropriation
(Plaintiff
Structural Dynamics Research Corp. v. Engineering Mechanics Research Corp.)
(Defendant
Amoco Production Co. v. Lindley)))

MOST-ON-POINT-NEAR-MISS-CASES-BY-CLAIM:
((Breach-Of-Contract (Plantiff)(Defendant))
(Breach-Nondisclose-Noncompete-Agreement (Plaintiff) (Defendant))
(Trade-Secrets-Misappropriation
(Plaintiff
Analogic Corp. v. Data Translation, Inc.
Space Aero Products Co. v. R.E. Darling Co.
Telex Corp. v. IBM Corp. (1)
Structural Dynamics Research Corp. v. Engineering Mechanics Research Corp.)
(Defendant
Crown Industries Inc. v. Kawneer Co.
Midland-Ross Corp. v. Yokana
Automated-Systems v. Service-Bureau
Midland-Ross Corp. v. Sunbeam Equipment Corp.)))

BEST-CASES-BY-CLAIM:
((Breach-Of-Contract
(Plaintiff Laff v. John O. Butler Co.)
(Defendant))
(Breach-Nondisclose-Noncompete-Agreement
(Plaintiff
Structural Dynamics Research Corp. v. Engineering Mechanics Research Corp.)
(Defendant
Motorola, Inc. v. Fairchild Camera and Instrument Corp.))
(Trade-Secrets-Misappropriation
(Plaintiff
Structural Dynamics Research Corp. v. Engineering Mechanics Research Corp.)
(Defendant
Amoco Production Co. v. Lindley)))

POINTS:
(Arg-Rec-4 Arg-Rec-3 Arg-Rec-2 Arg-Rec-1 Arg-Rec-0)