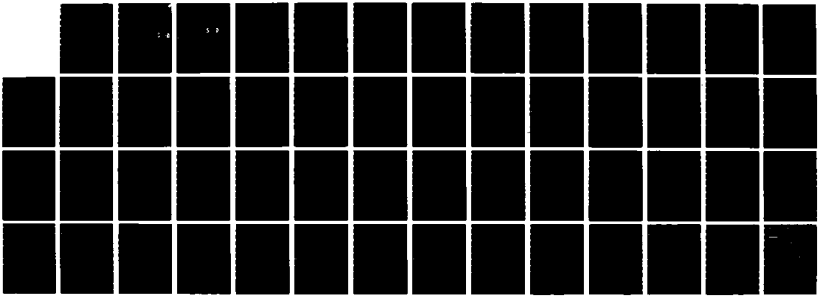
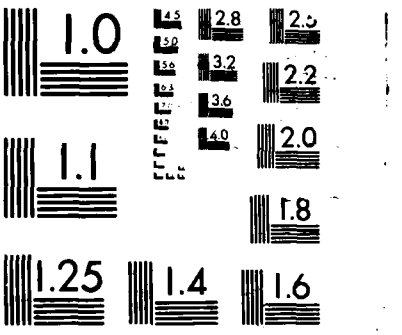
AD-A175 925 MODELLING LEGAL ARGUMENT: REASONING WITH CASES AND HYPOTHETICALS A THESIS..(U) MASSACHUSETTS UNIV AMHERST DEPT OF COMPUTER AND INFORMATION S.. K D ASHLEY 1/1

UNCLASSIFIED 30 JUN 86 CPTM-10 N00014-85-K-0017 F/G 5/7 NL




1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.8
3.2
3.6
4.0

AD-A175 925

The Counselor Project

Department of Computer and Information Science
University of Massachusetts
Amherst, Mass. 01003

DTIC
ELECTE
JAN 12 1987
S D

Modelling Legal Argument: Reasoning with Cases and Hypotheticals A Thesis Proposal

Kevin D. Ashley
June 30, 1986
CPTM #10

DTIC FILE COPY

DISTRIBUTION STATEMENT A
Approved for public release
Distribution Unlimited

This series of internal memos describes research in artificial intelligence conducted under DARPA Contract N00014-85-K-0017

86 10 28 062

(2)

DTIC
ELECTE
JAN 1 2 1987
S D
D

DISTRIBUTION STATEMENT A
Approved for public release
Distribution Unlimited

Modelling Legal Argument: Reasoning with Cases and Hypotheticals

A Thesis Proposal

Kevin D. Ashley

June 30, 1986

CPTM #10

Modelling Legal Argument: Reasoning with Cases and Hypotheticals A Thesis Proposal*†

Kevin D. Ashley‡
Department of Computer and Information Science
University of Massachusetts
Amherst, Massachusetts 01002

June 30, 1986

DTIC
COPY
INSPECTED

Accesion For
NTIS CRA&I ☑
DTIC TAB ☐
Unannounced ☐
Justification .....
By ltr on file
Distribution /
Availability Codes
Dist | Avail a. d / or Special
A-1

*This work was supported (in part) by: Grant IST-8212236 of the National Science Foundation, the Advanced Research Projects Agency of the Department of Defense, monitored by the Office of Naval Research under contract no. N00014-85-K-0017, and an IBM Graduate Student Fellowship.

†This thesis proposal has been prepared under the guidance of Edwina L. Rissland, Chairperson, Thesis Committee, in partial fulfillment of the requirements of Doctor of Philosophy, Computer and Information Science Department, University of Massachusetts

‡Copyright ©1986. Kevin D. Ashley. All rights reserved.

Contents

1. Introduction 4

2. Examples of How Experts Reason with Cases and Hypotheticals 5

3. Advantages of Case-Based Reasoning 8

4. Statement of Problem 9

5. Modelling Legal Argument 9

5.1 Cases, Claims, Holdings – Some Definitions 9

5.2 A Simplified Legal Argument 10

5.3 Responding to a Case Cited in an Argument 10

6. The HYPO Program 11

6.1 HYPO's Modules 13

6.2 The CASE-ANALYSIS Module 13

6.3 The ARGUMENT Module 16

6.3.1 Responding with a Competing Authority and by Distinguishing 20

6.3.2 Responding with a Hypothetical Counter-example 22

6.3.3 Conflict Examples 23

6.4 The FACT-GATHERING Module 24

6.5 The BEST CASE SELECTION Module 28

6.6 The EXPLANATION Module 30

6.7 The HYPO-GENERATOR Module 31

6.8 HYPO's Architecture 32

7. Expected Results and Discussion 35

7.1 How the Research Should be Evaluated 35

7.2 Assumptions and Simplifications 35

8. Related Work **37**

8.1 AI Models of Legal Reasoning 37

8.2 Argumentation 38

8.3 Memory Organization and Case-Based Reasoning 39

8.4 Explanation 40

8.5 Reasoning with Examples 41

8.6 Analogical Reasoning 41

8.7 Learning 42

9. Conclusion **44**

1. Introduction

In this thesis proposal, we outline a program, **HYPO**, to model reasoning with cases and hypotheticals. The program comprises a means of representing and indexing cases in a Case Knowledge Base ("CKB"), a computational definition of relevance in terms of "dimensions" which capture the utility of a case for making a particular kind of argument, a dimension-based method for comparing cases, and methods for selecting the best case for use in an argument, for distinguishing other cases, for making and responding to points in an argument, for asking pertinent questions and for generating hypotheticals based on seed cases in order to help an arguer formulate an argument, gather relevant facts, and explain his argument. [1]

HYPO's domain is legal argument where, as illustrated below with examples of oral arguments before the Supreme Court, experts use cases and hypotheticals as primary tools for making arguments. A real case is a case that has been litigated and decided; a hypo has not (even though it may be a very slight variation of one that has or may have been "engineered" to present some interesting issue not yet faced in a real case.).

After describing the advantages of Case-Based Reasoning ("CBR") over solely rule-based techniques, the proposal introduces the components of HYPO and works through examples of the tasks performed by the various modules of the program:

- **CASE-ANALYSIS** for finding relevant cases;
- **ARGUMENT** for generating and responding to legal points about a fact situation;
- **FACT-GATHERING** to ask sensible questions about additional facts;
- **BEST CASE SELECTION** for picking the best case in favor of a party's position;
- **EXPLANATION** to give reasons using examples for argument exchanges and factual queries; and
- **HYPO-GENERATOR** to make hypothetical fact situations tailored to the needs of argument making and explanation.

The paper presents a discussion of how the research should be evaluated, compares how other AI researchers have modeled legal reasoning and describes the relation of recent work in argumentation, case-based reasoning, dynamic memory organization, explanation, learning and analogical reasoning to the proposed thesis research.

---

[1] HYPO is embedded in a larger program called COUNSELOR which emphasizes natural language understanding, generation and discourse, in the domain of strategic resource allocation.

2. Examples of How Experts Reason with Cases and Hypotheticals

Experts reason with cases and hypotheticals in many fields such as mathematics, medicine, and business management, but particularly in law and especially in legal argument.

The uses that attorneys make of cases and hypotheticals as tools in argument are illustrated in the oral arguments before the United States Supreme Court. To the chagrin of counsel before the bar of the Supreme Court, the Justices frequently interrupt an attorney's presentation to pose hypotheticals. For example, in *Lynch v. Donnelly*, a case involving the constitutionality of the Christmas creche display of the City of Pawtucket, Rhode Island, Justices posed the following hypotheticals:

> To the attorney for the City:
>
> Q: Do you think ... that a city could display a nativity scene alone without other displays such as Santa Claus and Christmas trees...?
>
> Q: [C]ould the city display a cross for the celebration of Easter, under your view?
>
> To the attorney opposing the display:
>
> Q: [S]upposing the creche were just one ornament on the Christmas tree and you could hardly see it unless you looked very closely, would that be illegal?
>
> Q: What if they had three wisemen and a star in one exhibit, say? Would that be enough? ... What if you had an exhibit that had not the creche itself, but just three camels out in the desert and a star up in the sky?
>
> Q: Well, the city could not display religious paintings or artifacts in its museum under your theory.
>
> Q: There is nothing self-explanatory about a creche to somebody ... who has never been exposed to the Christian religion.
>
> Q: Would the display up on the frieze in this courtroom of the Ten Commandments be unconstitutional then, in your view?
>
> Q: Several years ago ... there was a ceremony held on the Mall, which is federal property of course. ...[T]here were 200,000 or 300,000 people ... and the ceremony was presided over by Pope John Paul II. Would you say that was a step toward an establishment of religion violative of the religion clauses? ... Then you think it would be all right to put a creche over on the Mall? ... How do you distinguish a high mass from a creche? ... [T]here was a considerable involvement of government in that ceremony, hundreds of extra policemen on duty, streets closed for traffic control purposes, and all that sort of thing. That was a considerable governmental involvement, was it not? [*SUP Lynch v. Donnelly*, Case No. 82-1256, Fiche No. 5, pages 9,11,32,37 45].

In the above questions, one can see the Justices modifying the fact situation along various dimensions: changing the location, focus, size, and symbolic religious content of the display, the nature of the viewer, and the degree of government involvement. Sometimes the purpose of the modifications is to compare the fact situation to actual cases previously decided by the court to test whether the current situation presents stronger or weaker facts. [2] Or the actual "case", like the Mall example, may be significant because it did *not* give rise to litigation.

Frequently, the Justices use the hypothetical to apply pressure to the rule proposed by an attorney for deciding the case. That can be seen in the Mall example above and in the following example from *New Jersey v. T.L.O.*, a case involving the constitutionality of a high school vice principal's search of a female student's handbag for cigarettes after a teacher reported that she had been smoking in the girls room. A Justice asked:

> Q: Do you think then that a male teacher could conduct a pat-down search of a young woman at age sixteen to find the cigarettes?

In response, the attorney for the state took the position that the Fourth Amendment of the United States Constitution, which has been interpreted as prohibitting unreasonable searches by law enforcement authorities, does not apply to high school administrators. The Justice rejoined:

> Q: And does that mean that their authority then to make searches, if the Fourth Amendment is completely inapplicable, extends to any kind of search, strip search or otherwise? [*SUP New Jersey v. T.L.O.*, 1984 Term, Fiche No. 5, pages 13 22].

Similarly, in *Sony Corp. v. Universal City Studios* a Justice posed the following series of hypotheticals to back an attorney into a corner. The attorney's position was that if Sony sold video recorders knowing that consumers would use them to copy copyrighted materials then Sony would be legally responsible to the owners of the copyrights. A Justice asked,

> Q: Suppose ... that about ten percent of all programming could be copied without any interference by the producer or whoever owned the program....
>
> A: I don't think that would make any difference. I think ten percent is too small of an amount.
>
> Q: Well, what about 50?

[2] See e.g., *Stone v. Graham*, 449 U.S. 39 (1980): Posting copies of Ten Commandments in schools held unconstitutional; *Gilfillan v. City of Philadelphia*, 637 F. 2d 924 (CA3, 1980): City-financed platform and cross used by Pope John Paul II to celebrate public mass held unconstitutional; *McCreary v. Stone*, 575 F.Supp. 1112 (SDNY 1983): Not unconstitutional for village not to refuse permit to private group to erect creche in public park.

The attorney for the Studios asserted that even if there were only one television program that was copyrighted, if Sony knew the program would be copied, it would be legally responsible. Finally, the Justice asked:

> Q: Under your test, supposing somebody tells the Xerox people that there are people who are making illegal copies with their machine and they know it. ... But your view of the law is that as long as Xerox knows that there's some illegal copying going on, Xerox is a contributory infringer?
>
> A: To be consistent, Your Honor, I'd have to say yes.
>
> Q: A rather extreme position. [*SUP Sony Corp. v. Universal City Studios*, Case No. 81-1687, Fiche No. 2, pages 21–25].

In the last two examples, although the altered fact situations posed by the Justice are still covered by the proposed rule, it is progressively harder for the attorney to justify applying the rule to the hypotheticals because the latter present progressively weaker facts. The Justice "stacks" the hypothetical with more extreme facts that weigh against the party in the hypothetical who corresponds to the attorneys client. The attorney is forced to distinguish the hypothetical, to come up with some alternative explanation for why the hypothetical and the current fact situation need not be decided the same way. This is especially true if the hypotheticals are closely based on real cases that the Court has decided, or for which there are strong reasons to decide, contrary to the way it would be decided if the proposed rule were followed.

To summarize, the above examples show how cases and hypotheticals are used as rhetorical tools in argument:

- To present, support and attack positions in an argument (e.g., by testing consequences of a tentative conclusion, pressing an assertion to its limits and exploring the meaning of a concept.)
- To relate a fact situation to significant cases from past experience.
- To factor a complex situation into component parts (e.g., by exaggerating strengths, weaknesses or by hypothetically eliminating features.)
- To control the course of an argument (e.g., by focusing attention of participants in a discussion on particular issues.)

3. Advantages of Case-Based Reasoning

A robust ability to reason with cases and hypotheticals is a powerful computational tool offering some important advantages over solely rule-based approaches:

When the rules run out or conflict... In some domains, although there are rules, the predicates of the rules are not sufficiently well defined to decide if the rule applies to a fact situation. This is particularly a problem if the domain does not have a strong underlying causal theory. In domains where there are more than one expert, there is lack of agreement about the rules. Some rules are inconsistent. If there are cases from which the rules have been inferred or in which experts have applied the rules, those cases may be used to analyze problem situations despite the lack of definitions for predicates or consensus about rules. One argues directly from the cases by emphasizing the relevant similarities and differences between the analyzed cases and the problem situation.

When the rules rationalize rather than analyze... In many domains where argument and persuasion are important, the expertise is not in using the rules to solve a problem. Rather, having solved a problem by whatever means, the task is to make it look as though the answer was deductively inferrable from the rules. In such domains, although experts' rules may not be capable of leading to the solution in the first place, case-based techniques may.

Explaining with Examples: Examples have persuasive power. Even if an expert explains his analysis of a problem by working through a trace of the rule-based inferences that can be applied to deduce the answer, the explanation can be improved by providing an example of a relevant real case susceptible to a similar analysis. Another strength of Case-Based Reasoning is that it provides a way of explaining **why not** to take a course of action by illustrating the negative consequences with an example.

A kind of behavior we expect from human experts. Not only do we expect experts to explain with examples we expect them to pose hypotheticals (i.e., worst cases, borderline cases, ideal cases, extreme cases.) Facility in manipulating hypotheticals is a measure of our confidence in an expert's understanding and a tool for planning and anticipating problems.

Learning from experience. Cases are a repository of experience. Rules are one means of accessing that experience but not the only or the best means. Accessing cases based on similarity and relevance is also viable. In addition, cases are extensional definitions of rules and more fully represent the experience of the expert.

4. Statement of Problem

The general problem of this research is to develop a program that can reason with cases and hypotheticals. One needs a task domain in which cases can be used to achieve a purpose, a computational definition of relevance in terms of a case's utility for performing the task, the ability to represent cases and to index and retrieve them according to relevance, a general method for comparing cases' utility and explaining why alternatives are less useful, a general method of generating meaningful hypothetical cases and a control structure that can apply the cases and hypotheticals to perform the task.

The legal domain is a good one for experimenting with meeting these requirements. The cases in a sufficiently narrow legal domain are stereotypical enough to be represented and indexed effectively. Legal cases serve a central function in the task of legal argument. Legal argument has some game-like features which make it possible to compare cases symbolically in terms of their utility in making or responding to an argument. The cases can be organized in terms of their utility in a manner that both facilitates their retrieval and the generation of hypotheticals. The discourse and game-like features of legal argument suggest the nature of the control structures to use.

The specific problem of this research, then, is to develop a program that can reason with cases and hypotheticals in the legal domain. Given a fact situation, for example, how can the program find relevant cases, select the stronger cases to make legal points about the fact situation and respond to those points by distinguishing the cited cases or by generating realistic hypotheticals to attack the points? How can the program explain those points and responses in a manner acceptable to attorneys? How can the program demonstrate its understanding of the fact situation by posing "What if..." hypotheticals based on the fact situation and by asking intelligent questions about additional facts that may be significant?

5. Modelling Legal Argument

5.1 Cases, Claims, Holdings Some Definitions

In order to model legal argument, various kinds of domain knowledge need to be represented. Legal *cases* are disputes between parties tried by a court whose decisions are reported in published opinions. The opinion sets forth the facts of the case, the claims made by one party against the other, and the court's holding. The *facts* of the case are statements about events associated with the dispute that were proved at trial or which the court assumed to be true. A *claim* is a recognized kind of complaint for which courts will grant relief. Examples of claims are breach of contract, negligence, trade secrets misappropriation, or copyright infringement. The *elements* of a claim are generalized statements of what facts must be proven in order to prevail on the claim. For example, there are said to be three elements as a condition of the existence of a trade secret: "novelty, secrecy and value in the trade or business of the putative trade secret owner" [Gilburne & Johnston, 1982,

p. 215]. The *holding* is the decision of the court as to the legal effect on each claim of the facts of the case, either in favor of the plaintiff or the defendant.

5.2 A Simplified Legal Argument

A point in a legal argument can be as simple as an attorney's citing a prior decided case in favor of his position. The cited case usually involves some collection of facts that are similar in a relevant sense to those involving his client and has been decided in favor of a party whose position is comparable to that of the client.

In citing the case, the attorney, in effect, argues that the case stands for a rule, **the rule of the case**, and that if that rule is followed, the current fact situation should be decided in the same way as the cited case. The rule of the case can be thought of as a generalized holding. The rule states that if certain facts are true in a subsequent fact situation as they were in the cited case, then the court should decide a particular claim in favor of the plaintiff (or defendant) just as it did in the cited case.

5.3 Responding to a Case Cited in an Argument

There are at least six ways to respond to a case cited in an argument. To introduce them consider a simplified example of a point and response. Assume that an attorney representing the defendant cites a prior case that has been decided in favor of the defendant on a claim for misappropriation of trade secrets. Suppose that the only facts *shared* by the current fact situation and the cited case were that the plaintiffs in each were corporations. Since there are no other shared facts, citing the prior case in favor of the defendant's defense against the trade secrets claim in the current fact situation is, in effect, citing it for the rule that if the plaintiff is a corporation, then plaintiff should lose.

Plaintiff's attorney can respond to this specious legal argument in the following ways:

1. **Cite a competing authority**. The simplest and perhaps the best response is to come up with a counter-example, a real case where a corporate plaintiff won a claim for trade secrets misappropriation.

2. **Distinguish** the cited case by offering an **alternative explanation** of the cited case. The attorney points out some other facts that were true of the cited case but not of the current fact situation, that strongly favored the defendant and that provide an alternative explanation of why the plaintiff lost in the cited case, circumventing and negating the conclusion that all corporate plaintiff's lose.

3. Distinguish the cited case by showing **exceptional circumstances** in the current fact situation. This is similar to the previous method but the focus is on finding facts in the current

fact situation that are not present in the cited case and that so strongly favor the plaintiff that, even if corporate plaintiff's usually lose, this corporate plaintiff should win.

4. Cite a **hypothetical** counter-example. The attorney generates a counter-example, a hypothetical case where the facts so strongly favor the corporate plaintiff that the plaintiff should win.

5. Cite a **statutory** provision. The attorney cites a provision of an applicable statute or constitution from which it can be inferred that this plaintiff should win.

6. Make a **policy** argument. The attorney explains, for example, that in our legal system, the fact that a party is a corporation does not usually count as a reason why it should not win a lawsuit, that such a rule would unfairly discriminate against corporations whose status is protected under various statutes and the Constitution, etc.

7. Attack the cited case's **pedigree**. The attorney argues that the court need not follow the cited case because, for example, the case was not decided by a court in the chain of command of appellate courts of the jurisdiction.

As shown in Section 6.3, even if the cited case and the subsequent fact situation shared more facts that were relevant to the merits of plaintiff's trade secrets claim, the same modes of response would be employed.

6. The HYPO Program

HYPO is a program to model reasoning with cases and hypotheticals ("hypos"). The program comprises a means of representing and indexing cases in a Case Knowledge Base, a computational definition of relevance in terms of "dimensions" which capture the utility of a case for making a particular kind of argument, a dimension-based method for comparing cases, and methods for generating legally meaningful hypotheticals based on seed cases.

For the thesis, the program will be extended to enable it to reason well enough to perform the first four kinds of responses described above (i.e., citing a competing authority, distinguishing via alternative explanations or exceptions, citing a hypothetical counter-example), [3] to generate point/response exchanges, to ask pertinent questions about the fact situation, to select the best

[3]The fifth kind of response, citing rules from statutes, regulations or constitutions from which a lawyer claims the court can infer how it should decide an issue, is, of course, important. The rules, however, rarely are so well-defined that they can be applied to a current problem without further analysis involving referring back to prior cases where the rules have been applied in concrete fact situations. For this analysis, the first four kinds of responses become crucial.

The sixth kind of response, making a policy argument, involves abstract reasoning about facts and legal concepts and is more difficult. It remains to be seen what insights about this mode of response can be gained by implementing

Telex Corp. v. IBM Corp., 510 F.2d 894 (5th Cir., 1975).
Held for plaintiff IBM on trade secrets misappropriation claim where Telex gained access to IBM's confidential product development information by hiring an IBM employee, paying him a large bonus to develop a competing product. The employee used development notes he brought from IBM. Telex saved time and expense developing the competing product.

Midland-Ross Corp. v. Sunbeam Equipment Corp., 316 F.Supp. 171 (W.D. Pa., 1970).
Held for defendant Sunbeam on trade secrets misappropriation claim where Midland-Ross disclosed it's technical product development info to 100 persons.

Data General Corp. v. Digital Computer Controls, Inc., 357 A.2d 105 (Del. Ch. 1975).
Held for plaintiff Data General on trade secrets misappropriation claim where Data General disclosed its technical product development info to 6000 persons, all of whom were subject to nondisclosure agreements.

Automated Systems, Inc. v. Service Bureau Corp., 401 F.2d 619 (10th Cir., 1968).
Held for defendant SBC on trade secrets misappropriation claim where Automated-Systems' confidential info was about customer's business operations.

Table 1: Sample Cases from Case Knowledge Base.

case to cite in support of a party's position and to explain its questions and points using case examples and hypos.

In HYPO, cases are represented by a hierarchical cluster of frames (flavor instances) with slots for relevant features (plaintiff, defendant, claim, facts, etc.) [Ashley & Rissland, 1985; Rissland, Valcarce & Ashley, 1984]. Some features are in turn expanded and represented as frames (e.g., plaintiff). In dealing with a new fact situation, HYPO begins with a representation of the facts pre-processed into its frame representation.

HYPO's current CKB contains a dozen or so of the leading cases for trade secret law. A trade secret case frequently involves two corporations, a plaintiff and defendant, who produce competing products. The plaintiff usually alleges that the defendant gained an unfair competitive advantage in developing and marketing its product by misappropriating trade secret information developed by the plaintiff. See Table 1 for a sampling of trade secrets cases. A detailed example of HYPO's representation of a hypothetical case can be found in [Rissland, Valcarce & Ashley, 1984].

---

the first four. The seventh mode, attacking a case's pedigree, is too domain dependent, and not sufficiently interesting, to undertake at this time.

6.1 HYPO's Modules

The program has the following modules:

1. **CASE-ANALYSIS:** to analyze a fact situation to determine what cases are relevant to the situation, to retrieve relevant cases and organize them according to how they help or hurt the parties in the fact situation;
2. **ARGUMENT:** to make a legal point in connection with a fact situation and respond to it;
3. **FACT-GATHERING:** to guide questioning a user about the salient features of a fact situation for the purpose of providing a legal analysis;
4. **BEST CASE SELECTION:** to select the best case that supports a party's position.
5. **EXPLANATION:** to explain a query, point or response using examples;
6. **HYPO-GENERATOR:** to make hypothetical fact situations according to specifications provided by the ARGUMENT, FACT-GATHERING or EXPLANATION modules.

The modules are described in more detail in Sections 6.2 through 6.7 The program's flow of control is discussed in Section 6.8

6.2 The CASE-ANALYSIS Module

Besides the CKB and the understanding of the legal domain that this case representation implicitly contains, the other major source of domain-specific legal knowledge is in HYPO's ***dimensions***. Dimensions capture the notion of legal relevance of a cluster of facts to the merits of a claim: that is, for a particular kind of case, what collections of facts represent ***strengths*** and ***weaknesses*** in a party's position. The short answer is that facts are relevant to a claim if there is a court that decided such a claim in a real case and expressly noted the presence or absence of such facts in its opinion. Examples of dimensions in HYPO's area of trade secret law are: *Secrets-voluntarily-disclosed*, *Disclosure-subject-to-restriction*, *Competitive-advantage-gained*, *Vertical-knowledge*. These dimensions are summarized in Table 2. There are about thirty dimensions in all. The dimensions were gleaned from law journal articles describing the state of the (case) law in this area [Gilburne & Johnston, 1982].

As the samples indicate, each dimension has several facets:

1. **Prerequisites**
2. **Focal-slots**
3. **Ranges**
4. **Direction-to-strengthen-plaintiff**
5. **Significance**
6. **Cases-indexed**

For instance, the prerequisites of the *Secrets-voluntarily-disclosed* dimension, Table 2, are that two corporations, plaintiff and defendant, compete with respect to a product, plaintiff has confidential product information to which defendant has gained access and plaintiff has made some disclosures of the information to outsiders. The prerequisites are stated in terms of factual predicates, which indicate the presence or absence of a legal fact or attribute (e.g., existence of a product, existence of a non-disclosure agreement). The focal slot of this dimension is the number of disclosees and its range is a non-negative integer. To strengthen the plaintiff's position in a fact situation to which this dimension applies, decrease the number of disclosees; the best case being that with 0 disclosees. The significance of the dimension is that courts have found that the prerequisite facts are a reason for deciding a trade secrets misappropriation claim. This dimension indexes at least two cases in the CKB: *Midland-Ross* in which the court held for the defendant where the plaintiff disclosed the secret to 100 persons, [4] and *Data-General* in which the court held for plaintiff where plaintiff disclosed to 6000 persons. [5]

Dimensions facilitate looking at the same basic facts from different perspectives. For example, the fact that a corporation with a trade secret entered into a nondisclosure agreement with its former employee who subsequently worked for a competing corporation can be viewed from the perspectives of a trade secrets misappropriation claim or a contracts claim. The former focusses on the fact that by virtue of entering the nondisclosure agreement, the employee is more likely to know that information he learned on the job was confidential. The latter focusses on the whether the agreement is enforceable and whether by working for the competitor the employee breached the agreement. Different dimensions and the cases from which they are derived capture both viewpoints.

As shown in Figure 1, the CASE-ANALYSIS module takes a fact situation, determines which dimensions apply and which are near-misses and retrieves the cases in the CKB that are relevant to

[4] *Midland-Ross Corp. v. Sunbeam Equipment Corp.*, 316 F.Supp. 171 (W.D. Pa., 1970), Table 1

[5] *Data General Corp. v. Digital Computer Controls, Inc.*, 357 A.2d 105 (Del. Ch. 1975), Table 1

Secrets-voluntarily-disclosed:

Significance: Plaintiff's (P's) position stronger the fewer persons to whom secrets disclosed.
Prerequisites: P and Defendant (D) compete; D had access to P's product information; some disclosures.
Focal slot: Number of disclosees.
To Strengthen P: Decrease number of disclosees.
Range: 0 to N.
Cases indexed: *Midland-Ross, Data-General*

Disclosures-subject-to-restriction:

Significance: P's position stronger the fewer disclosees not subject to nondisclosure agreements.
Prerequisites: Competition; access to info; some disclosures and nondisclosure agreements.
Focal slot: Number of disclosees subject to restriction.
To Strengthen P: Increase percentage of disclosees subject to restriction.
Range: 0 - 100 %.
Cases indexed: *Data-General*

Competitive-advantage-gained:

Significance: P's position stronger the greater competitive advantage gained by D.
Prerequisites: Competition; access to info; D saved some expense.
Focal slot: Development expense saved.
To Strengthen P: Increase expense saved by D.
Range: 0 - 100 %.
Cases indexed: *Telex v. IBM*

Vertical-knowledge:

Significance: P's position stronger if information technical, not vertical.
Prerequisites: P and D compete; D had access to P's product information; info about something.
Focal slot: What information is about.
To Strengthen P: Make information about technical development of product.
Range: {technical, vertical}
Cases indexed: *Automated Systems*, et al.

Table 2: Sample Dimensions.

the fact situation. The output of the CASE-ANALYSIS module is the *Case-analysis-record* which contains:

1. applicable factual predicates;
2. applicable dimensions;
3. near-miss dimensions;
4. applicable claims;
5. relevant cases in the CKB;
6. points-and-responses.

As a first approximation, the cases indexed by the applicable dimensions are **relevant** to the fact situation. **Near-miss** dimensions are those for which the particular prerequisite associated with the focal-slot is the only prerequisite left to be satisfied by the fact situation. In *secrets-voluntarily-disclosed*, for example, the prerequisite associated with the focal-slot, number of disclosees, is whether the plaintiff made some disciosures.

6.3 The ARGUMENT Module

Having analyzed a fact situation and prepared a case-analysis-record, HYPO can generate fragments from a legal argument about the case. Alternatively taking the side of plaintiff and defendant, the program generates a 3-ply exchange consisting of a *point*, *response*, and *counter-response*. The program uses cases and made-up hypotheticals based on cases to generate the moves in the argument exchange.

An ARGUMENT module, depicted in Figures 2 and 3, generates the points and responses. To make a point, let us say for the plaintiff, the program must:

1. select the best case from the retrieved cases favorable to the plaintiff; and
2. abstract a rule from that case that applies to the current fact situation and leads to the conclusion that the plaintiff should win.

To make a response, the program must do as many of the following as possible:

1. Cite a competing authority (i.e., a real counter-example);
2. Distinguish by providing an alternative explanation of a cited case;

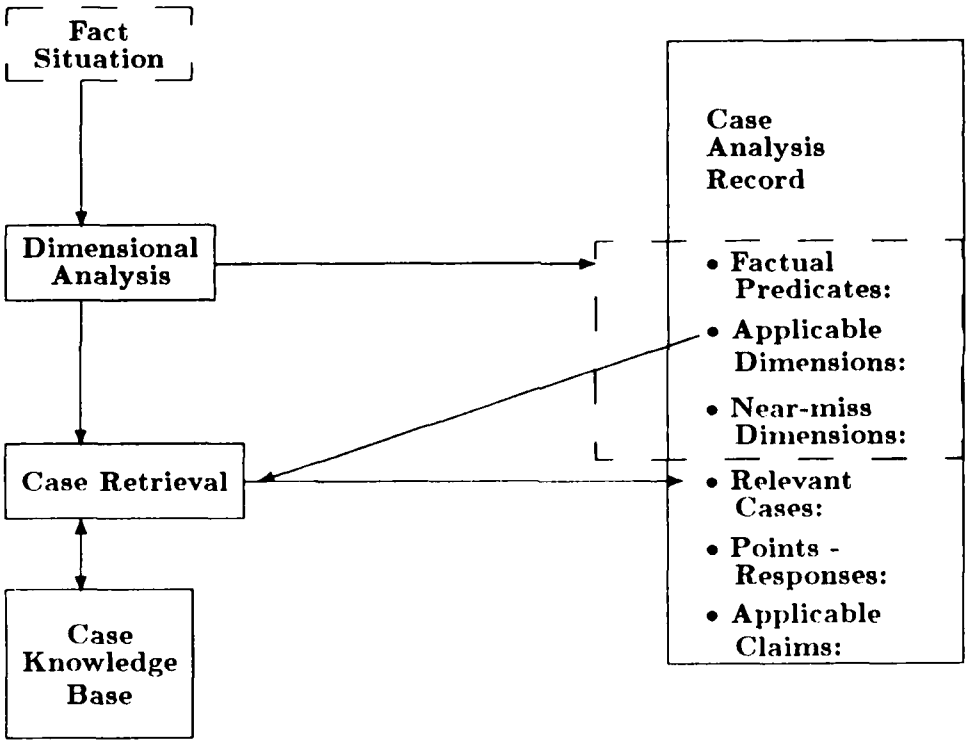


Figure 1: HYPO's CASE-ANALYSIS Module.

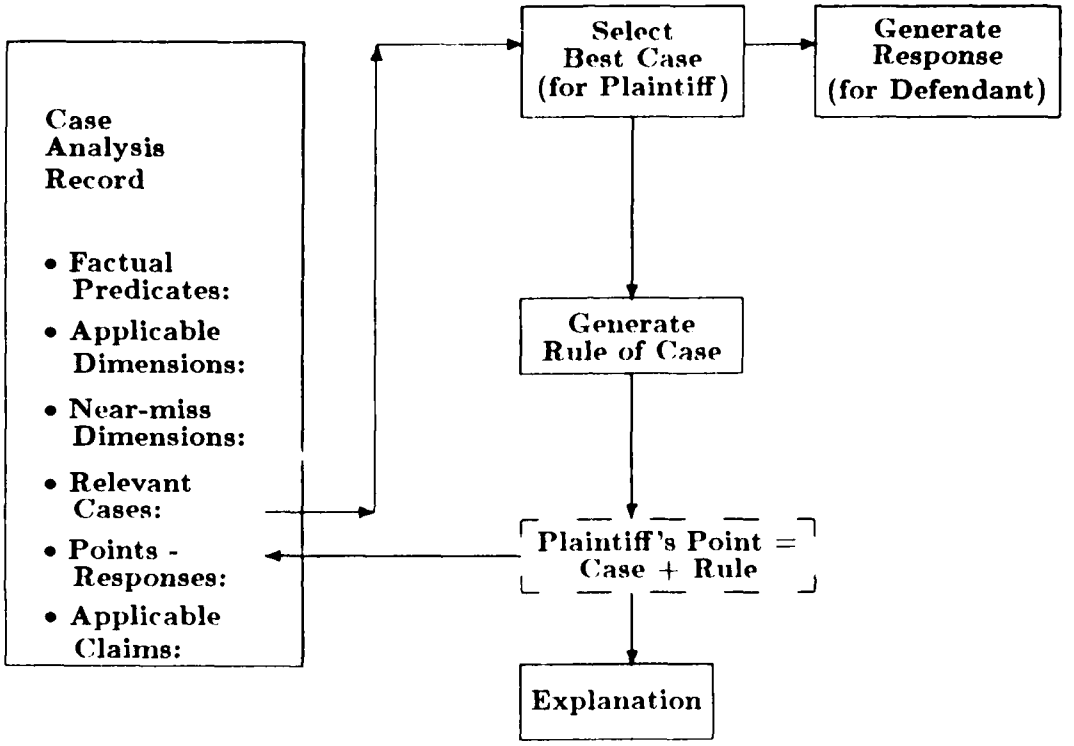


Figure 2: HYPO's ARGUMENT Module: Generating a Point (e.g., for Plaintiff.)

3. Distinguish by finding an exception to the rule of the cited case that exempts the current fact situation;

4. Attack a rule by citing hypothetical counter-examples.

The process of generating points and responses is illustrated in the following examples of argument tasks: *Responding with a Competing Authority and by Distinguishing*, 6.3.1, illustrating response methods 1 through 3, and *Responding with a Hypothetical Counter-example*, 6.3.2, illustrating response method 4.

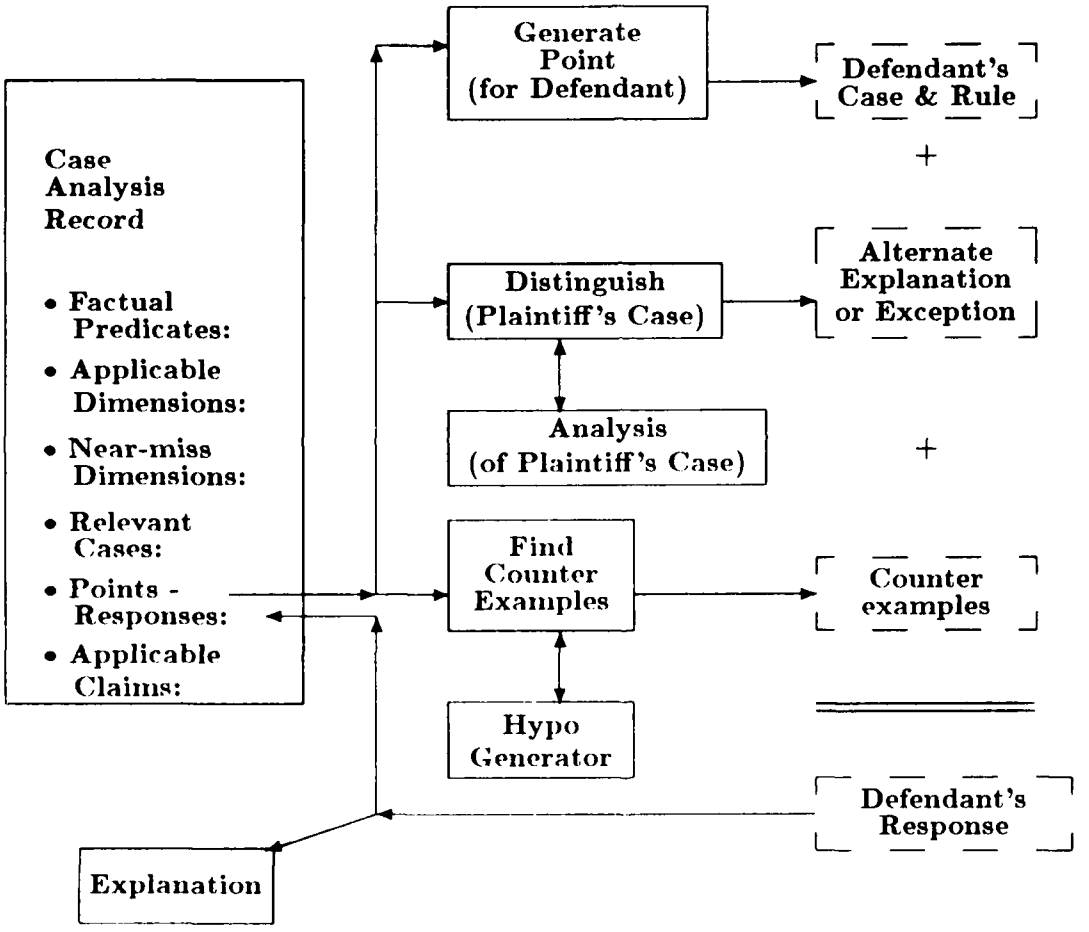


Figure 3: HYPO's ARGUMENT Module: Generating a Response (e.g., for Defendant.)

Widget-King, plaintiff v. *Cupcake*, defendant

Version 1:

Plaintiff Widget-King and Defendant Cupcake are corporations that make competing products. Widget-King had confidential information regarding its product. Cupcake gained access to Widget-King's confidential information and saved some time and expense developing its competing product. Widget-King had disclosed its information to 5 persons.

Version 2:

Plaintiff Widget-King and Defendant Cupcake are corporations that make competing products. Widget-King had confidential information regarding its product. Cupcake gained access to Widget-King's confidential information. The information is about customer's business methods.

Table 3: Sample Fact Situations.

6.3.1 Responding with a Competing Authority and by Distinguishing

The first example argument task involves Version 1 of the Sample Fact Situation presented in Table 3, Widget-King, plaintiff v. Cupcake, defendant. Cupcake has gained a competitive advantage by accessing confidential information about a Widget-King product. Widget-King has disclosed the information to five persons. The initial case-analysis-record for Version 1 of *Widget-King v. Cupcake* is:

applicable dimensions:
secrets-voluntarily-disclosed
competitive-advantage-gained
near-miss dimensions:
disclosures-subject-to-restriction
vertical-knowledge
relevant CKB cases:
Midland-Ross, Data-General

Given the current fact situation, the cases in the CKB shown in Table 1, and the task of arguing for the plaintiff, the program comes up with something like the following point/response exchange. Widget-King's point is: Since the plaintiff in the *Data-General* case won with 6000 disclosees, Widget-King should win with only 5. Cupcake's response is to point out that the plaintiff in the *Data-General* case won because all disclosees were subject to nondisclosure agreements (i.e.,

restricting them from spreading the secrets further) while none of Widget-King's disclosees were similarly restricted. Cupcake's response goes on to point out that in the *Midland-Ross* case, the defendant won where there were some disclosees, so Cupcake should win. Widget-King's counter-response is to point out that in *Midland-Ross*, 100 people had been disclosed the information, while Widget-King only disclosed to 5.

Briefly, the argument exchange example is generated in the following manner:

For Widget-King's point, the program determines that the prerequisites of the *secrets-voluntarily-disclosed* dimension, Table 2, are satisfied by the current fact situation and retrieves the *Data-General* case as a relevant, favorable case (i.e., the case is indexed by an applicable dimension and held for the plaintiff.) The program abstracts a rule from the *Data-General* case which applies to the current fact situation:

> *If* plaintiff and defendant corporations make competing products, defendant had access to plaintiff's confidential product information and the information was disclosed to 6000 or fewer persons,
>
> *then* hold for the plaintiff on a trade secrets claim.

The left hand side of the rule comes from the prerequisites and focal slot of the *secrets-voluntarily-disclosed* dimension, the dimension that lead to the relevant case; the "or fewer" comes from the range and to-strengthen-plaintiff slots of the *secrets-voluntarily-disclosed* dimension. The right hand side comes from the decision of the retrieved case (i.e., who won it and the kind of claim involved.)

In responding for Cupcake, the program notes that *Data-General* is strong on the *disclosures-subject-to-restriction* dimension, Table 2 It makes a new rule for *Data-General* based on this second dimension which is similar to the plaintiff's rule but includes the added condition: "and all disclosees subject to nondisclosure agreements." This modified rule provides an alternative explanation of the decision in the *Data-General* case but does not apply to the current fact situation. The program also retrieves a competing authority, the *Midland-Ross* case, a pro-defendant case indexed by the *secrets-voluntarily-disclosed* dimension, and uses it to fashion a rule to the effect that if there are disclosures, defendant wins. Note that the *Midland-Ross* case is a counter-example to the plaintiff's rule; it is a case involving fewer than 6000 disclosees where the defendant won.

For a counter-response, the program notes from the range and to-strengthen-for-plaintiff slots of the *secrets-voluntarily-disclosed* dimension that fewer disclosees are better for plaintiff, so it distinguishes *Midland-Ross* by noting that 5 is much less than 100.

In summary, this example has illustrated making a point by abstracting the rule of the case and making responses by citing a competing authority (*Midland-Ross*), showing an alternative explanation (of *Data-General*), and showing an exception (to *Midland-Ross*).

6.3.2 Responding with a Hypothetical Counter-example

The next example argument task involves Version 2 of the Sample Fact Situation presented in Table 3. The initial case-analysis-record for Version 2 is:

applicable dimensions:
vertical-knowledge
near-miss dimensions:
secrets-voluntarily-disclosed
competitive-advantage-gained
relevant CKB cases:
Midland-Ross, Data-General

Version 2 is like the first fact situation except that there are no disclosures and the confidential information that Widget-King wants to protect is about customer's business information, that is, the knowledge is "vertical" knowledge. As an example of vertical knowledge, suppose a software company developed an inventory control program for an automobile parts dealer and sent its employee to spend a few months with the dealer to learn the dealer's methods for controlling and accounting for its inventory. That information would be called vertical knowledge by reference to the "vertical" relationship between a supplier and its customer. The court in *Automated-Systems* [6] decided that vertical information is not subject to trade secrets protection and held for the defendant.

Given the current fact situation, the cases in Table 1 and the task of arguing for the defendant, the program comes up with something like the following point/response exchange. Cupcake's point is: since the defendant in *Automated-Systems* won where the information was vertical knowledge, Cupcake should win. Widget-King's response is: Suppose in the *Telex* case, [7] that the information were vertical knowledge. If the rule of *Automated-Systems* were followed, plaintiff IBM would lose even though the information gave it a substantial competitive advantage and defendant bribed plaintiff's employee to gain the information.

Briefly, the argument exchange example is generated in the following manner:

For Cupcake's point, the program determines that the prerequisites of the *vertical-knowledge* dimension, Table 2, are satisfied by Version 2 and retrieves the *Automated-Systems* case which held for the defendant. The program abstracts a rule from *Automated-Systems* which applies to the

[6] *Automated Systems, Inc. v. Service Bureau Corp.*, 401 F.2d 619 (10th Cir., 1968), Table 1

[7] *Telex Corp. v. IBM Corp.*, 510 F.2d 894 (5th Cir., 1975), Table 1

current fact situation:

> *If* plaintiff and defendant corporations make competing products, defendant had access to plaintiff's confidential product information and the information was about customer's business methods,
>
> *then* hold for the defendant on a trade secrets claim.

In deciding how to respond on behalf of Widget-King, the program finds that it cannot distinguish the *Automated-Systems* case from the current fact situation along the *vertical-knowledge* dimension; there are no significant differences as one can see from Tables 2 and 1 This failure causes the program to look at fact situations that are farther afield from the current fact situation. The program tries the *competitive-advantage-gained* dimension of which most but not all of the prerequisites are satisfied (i.e., a near-miss dimension) and retrieves the *Telex* case.

Even though *Telex* held for the plaintiff, the program should not cite the case as a competing authority because, in generating the defendant's counter-response, the program could so easily distinguish *Telex*. That is, the program would point out that, whatever the rule in *Telex*, an exception should be made for Version 2 which involved vertical knowledge. The program would also point out the missing prerequisite, that *Telex* was so decided because the defendant had gained a substantial competitive advantage from the information and that there is no information in Version 2 that Cupcake gained a comparable advantage.

The program still has a use for the *Telex* case in this context, however. The program can attack the rule of *Automated-Systems* by posing a hypothetical where following the rule would lead to the wrong result. The hypothetical is based on the *Telex* case except that the information is about customer's business methods rather than technical information about the product. If the rule of *Automated-Systems* were followed, plaintiff would lose despite the close similarity of the case to *Telex* where the plaintiff won. The ARGUMENT module calls upon the HYPO-GENERATOR to construct this hypothetical. See Section 6.7.

The hypothetical based on *Telex* is a counter-example to the defendant's rule. Such a hypothetical is more effective the more it is "stacked" in favor of the plaintiff.[8] The more counter-examples there are, the more powerful plaintiff's attack on defendant's proposed rule.

6.3.3 Conflict Examples

In each of the argument tasks in Sections 6.3.1 and 6.3.2, the ARGUMENT module identified cases in which two dimensions conflict: the *Data-General* case and the hypothetical based on the

[8] By contrast, a real case counter-example is more powerful to the extent that it is not distinguishable by the defendant from the cited case. A hypothetical fact situation that is supposed to be a counter-example to the rule of a cited case but that is identical to the cited case would not be very convincing.

Telex case. These are called **conflict examples**. In each conflict example, the facts of the case make it strong for the plaintiff on one dimension and weak on another. Which ever way the court decided the case, for the plaintiff or the defendant, the case constitutes a weighting of the relative importance of the two reasons represented by the dimensions. Such a case is useful in resolving a dispute when the same conflicts arise in a subsequent fact situation. HYPO indexes conflict-examples as they are discovered or generated in the course of the ARGUMENT module's functions so that they can be accessed again. Discovering conflict examples in the course of argument is important because, as a practical matter, it is not always possible to build into the program *a priori* the knowledge that two particular dimensions conflict.

6.4 The FACT-GATHERING Module

Basically, what HYPO does is to start with a given fact situation and ask questions about possible facts that would make precedent cases apply in favor of or against various claims of the plaintiff. Since the space of possible facts known to the system is large, it needs to be searched heuristically so that the system asks only those questions which directly pertain to making an argument about the claim.

In effect, HYPO locates the fact situation in the CKB and incrementally moves out from it in various directions toward cases that would be useful for arguing in favor of a party. Dimensions provide the guidance for directing this search.

The process, depicted in Figure 4, takes place in the following steps:

1. Analyze the fact situation to identify the near-miss dimensions.
2. Select the near-misses that are most *pertinent* to the argument goal.
3. Ask questions to see if the near-miss dimension applies and to determine where the fact situation lies on the dimension.

The first step is performed by the CASE-ANALYSIS module and results in the case-analysis-record described in the last section.

The second step is performed by the FACT-GATHERING module. In essence, the near-miss dimensions are ordered according to their utility in satisfying the current argument goal of the user/system interaction. There are two possible goals:

1. To "build a case" in favor of a plaintiff's claim, by turning up facts upon which a strong pro-plaintiff precedent case involving the claim can be cited;
2. To advise the user of weaknesses, real or potential, by finding facts that the defendant could use to cite a strong pro-defendant case or to distinguish a pro-plaintiff case. In other words, to play Devil's Advocate.

As illustrated below in an extended example, each goal guides the FACT-GATHERING module in selecting near-miss dimensions to ask about. If the program needs to build a case in favor of a claim of the plaintiff, then it screens dimensions that index pro-plaintiff cases involving that claim. If it is playing Devil's Advocate, it (a) screens dimensions that index pro-defendant cases or (b) screens dimensions that lead to facts that distinguish the plaintiff's cases.

The third step is performed in part by the FACT-GATHERING module. The module initiates a question about the missing prerequisite of the selected near-miss dimension. [9] If the dimension is found to apply, the module inquires about the dimension's focal slot value in the fact situation. The program compares the focal slot value in the fact situation with that of a real case indexed by the dimension. The relative values of the focal slots determine whether the fact situation is stronger or weaker than the real case along that dimension and whether the case can be cited on behalf of a party's claim in the fact situation.

In order to illustrate this process, we will use Version 3 of *Widget-King v. Cupcake*, whose facts are as follows:

> Plaintiff Widget-King and defendant Cupcake are corporations that make competing products. Widget-King has confidential information concerning its own product. Cupcake gained access to Widget-King's confidential information. [10]

The parts of the initial case-analysis-record for *Widget-King v. Cupcake* that are relevant for the following discussion are:

applicable dimensions: nil
near-miss dimensions:
competitive-advantage-gained
secrets-voluntarily-disclosed
vertical-knowledge
relevant CKB cases: nil

The FACT-GATHERING module uses the goal and the kinds of cases indexed by the near-miss dimensions to select among them. Of the three near-miss dimensions, *competitive-advantage-gained* indexes only a pro-plaintiff case (i.e., a case that the plaintiff won), *vertical-knowledge* indexes only

[9] As part of the COUNSELOR project, the HYPO program has been designed to motivate the system to generate natural language questions and explanations by example and to understand the user's answers but it does not itself express or understand natural language outputs or inputs.

[10] Another aspect of fact gathering is guiding the user through the process of providing an initial description of the fact situation. In the COUNSELOR project, we are experimenting with using scripts about stereotypical legal disputes involving trade secrets and about attorneys' representing clients to guide the initial questioning and understanding.

a pro-defendant case and *secrets-voluntarily-disclosed* indexes both a pro-defendant and pro-plaintiff case. Although the number of cases in this example is very small, as the number of cases in the CKB increases, some dimensions will continue to index only cases that favor the plaintiff or the defendant, but not both. Others will index cases of both types.

If the goal is to build a favorable argument for plaintiff Widget-King, the program will select *competitive-advantage-gained* as the first dimension to ask about. This is the first choice because the dimension indexes only pro-plaintiff cases. The CKB contains no case in which the advantage conferred on the plaintiff by virtue of the defendant's having saved development expense was so overshadowed by facts favoring the defendant that the defendant won. If the goal were to build an argument for defendant, *secrets-volutarily-disclosed* would be the near-miss dimension of choice.

If *competitive-advantage-gained* is selected, the program first asks about the missing prerequisite, whether defendant Cupcake saved any time or expense developing its competing product. If so, the program asks how much time or expense had been saved. If the relative amount of the savings is comparable to or greater than that in *Telex*, that case can be cited in favor of plaintiff Widget-King.

The *secrets-voluntarily-disclosed* dimension is the program's second choice in trying to build an argument for the plaintiff. The program needs to find out if Widget-King disclosed its secrets to anyone and whether there were 6000 or fewer disclosures. If so, the *Data-General* case, with its pro-plaintiff holding despite 6000 disclosees can be cited in favor of Widget-King. The dimension is the second choice because it also indexes a pro-defendant case, *Midland-Ross*. In other words, participation in this dimension does not necessarily help the plaintiff. The same dimension might have been selected by the program acting on a goal to build an argument for the defendant. In that event, the program needs to determine if there are 100 or more disclosures. Then *Midland-Ross* could be cited on behalf of defendant Cupcake.

Assume that in response to its questions, the program is told that Widget-King made 110 disclosures. The new fact has a number of effects. The CASE-ANALYSIS module updates the case-analysis-record to reflect that *secrets-voluntarily-disclosed* is now an *applicable* dimension. *Data-General* and *Midland-Ross* are added as relevant cases. The former can be cited in favor of the plaintiff; the latter can be cited in favor of the defendant. Significantly, with the addition of disclosures to the fact situation, the *disclosures-subject-to-restriction* dimension becomes a *near-miss*.

Switching to Devil's Advocate mode, the program tries to poke holes in the plaintiff's argument by distinguishing the *Data-General* case. That is, it tries to find some fact that makes the *Data-General* case stronger for the plaintiff than the fact situation. There is such a fact. In *Data-General* all of the disclosees were subject to the restriction that they would not tell the secret to anyone else. The dimension *disclosures-subject-to-restriction* captures this information and indexes the *Data-General* case. If Widget-King's disclosees are not similarly restricted, *Data-General* does not really help Widget-King's argument; it is distinguishable.

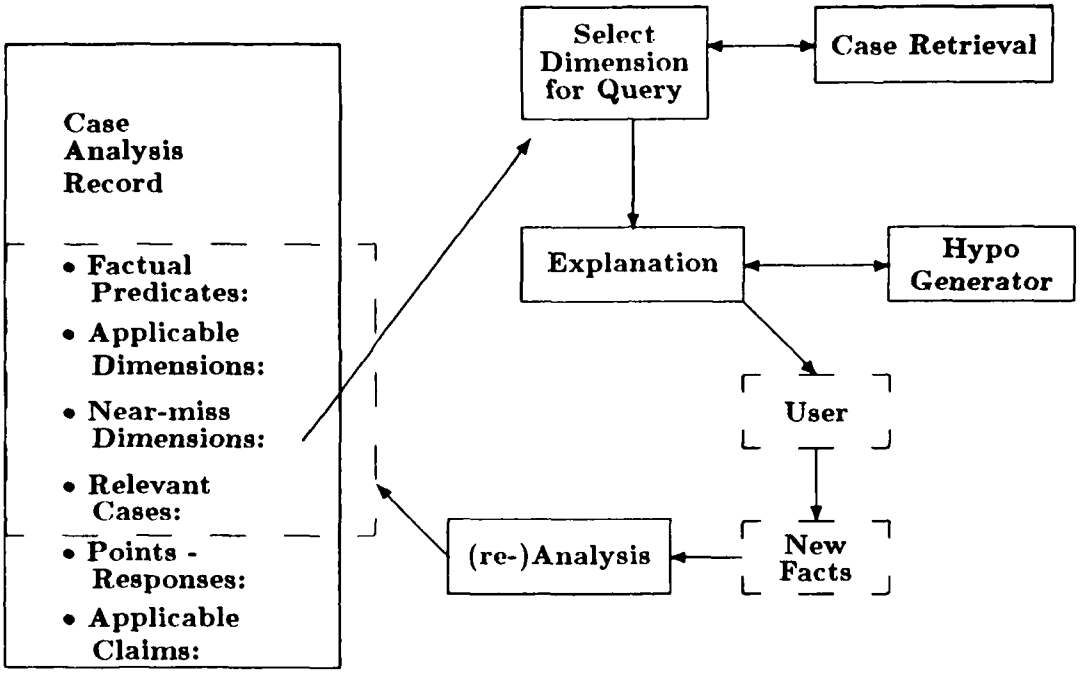


Figure 4: HYPO's FACT-GATHERING Module.

The need to distinguish *Data-General* prompts the FACT-GATHERING module to ask the user whether any of Widget-King's disclosures were subject to restriction and how many. Remember that the *disclosures-subject-to-restriction* dimension has just become a near-miss dimension and also applies to *Data-General*, the case the program is trying to distinguish. On that basis, the program asks about the dimension's missing prerequisite, whether any of Widget-King's disclosures are subject to restriction and if so how many. If the *Widget-King* case turns out to be weaker along *disclosures-subject-to-restriction*, or if that dimension does not apply at all because none of Widget-King's disclosures are subject to restriction, then HYPO has found a way to distinguish *Data-General*.

6.5 The BEST CASE SELECTION Module

Given a fact situation, real or hypothetical, HYPO must choose the case that is best for a particular party. The goal is to let the program make the strongest point it can (i.e., select a case so as to maximize the weakness of the response.) The choices arise because all of the facts may not have been specified, more than one dimension may wholly or partially apply to the current fact situation and each dimension may index a number of cases.

The best case for, let us say, a defendant on a trade secrets misappropriation claim is one that holds for the defendant on such a claim, shares the greatest number of common dimensions with the current fact situation, is not distinguishable by the plaintiff, and the rule of which as applied to the current fact situation is not assailable. The worst case is one that holds for the plaintiff, shares the greatest number of common dimensions with the current fact situation, is not distinguishable by the defendant, and the rule of which as applied to the current fact situation is not assailable. In between the extremes of best and worst case are relevant cases, pro(con) the defendant that plaintiff(defendant) can distinguish or for which the rules as applied to the current fact situation are assailable. As stated in the discussion of the ARGUMENT module, a case is distinguishable by the plaintiff if for a response, plaintiff either could show that there is an alternative explanation for why the case held for the defendant that does not apply to the current fact situation or that there is an exception that applies in favor of the plaintiff in the current fact situation but not in the case. The rule of the case is assailable if counter-examples, real or hypothetical can be provided.[11]

The selection process is similar to using a minimax procedure to search a game tree. The program must look ahead to determine the opponent's responses to each of its possible moves. Even without looking many ply ahead, the process will be computationally expensive, so it is necessary to focus the program's efforts on candidate moves that are most likely to succeed.

Selecting the best case is a three step process:

1. Analyzing the fact situation to generate the candidates.
2. Screening for candidate best cases: The program uses heuristics to screen cases that are the most likely candidates for best case.
3. Comparing responses to candidates: The program generates and compares responses to each of the candidates, selecting the candidate to which it can make the weakest response as the basis of the strongest point the program can make.

[11] A real counter-example is a third case to which the rule of the cited case applies but which holds in favor of the plaintiff. For example, the program's counter-response in the first example would have been much stronger if in distinguishing *Midland-Ross* on behalf of Widget-King as involving far more disclosees the program also had come up with a competing authority that held in favor of plaintiff where there were 5 disclosees and no nondisclosure agreements.

Step one is performed by the CASE-ANALYSIS module and results in a case-analysis-record in which the candidates are the pro-party relevant cases.

In Steps two and three, responses are ranked according to the extent they involve [in descending order of importance]: citing a competing authority, distinguishing by finding an alternative explanation, distinguishing by finding an exception, [12] and citing a hypothetical counter-example. Steps two and three differ in that the former relies on heuristics for estimating whether a case is distinguishable or whether there are counter-examples to a rule of a case. The second step actually generates the responses and compares.

For the step two screening process, the program "locates" the candidates, in the CKB and uses heuristics to estimate whether their neighbors can be used to fashion responses to them. As a heuristic for determining whether the case is distinguishable HYPO checks on what dimensions apply to the case and current fact situation. An alternative explanation is likely if the case strongly favors the defendant on a dimension that applies only to the case or more strongly favors the defendant on the common dimensions. An exception is likely if the situation strongly favors the plaintiff on a dimension that only applies to it and not to the case. As a heuristic for determining whether there are counter-examples, the program knows that real case counter-examples are indexed under one or more of the dimensions that make the cited case relevant. Counter-examples are recognized because their holdings are contrary to the cited case eventhough they lie in a more favorable part of the dimension's range. The program also knows that hypothetical counter-examples can be fashioned from pro-plaintiff cases that are strong along near-miss dimensions that partially apply to the cited case.

Having selected the candidate cases, the second step involves actually generating and comparing responses to each. This step is necessary to make finer comparisons of the candidate cases than the heuristics allow and to catch any cases with *pitfalls* (i.e., cases that support the party's position on one ground but have facts that detract from it on another.)

An example of a hard comparison of responses is presented in the first argument exchange example. In deciding how to make the plaintiff's point, the program actually has to decide between two pro-plaintiff relevant cases, the *Data-General* case and the *Telex* case.[13] The program's response to citing the *Data-General* case on behalf of the plaintiff combined a real counter-example, the *Midland-Ross* case, and an alternative explanation of *Data-General*. In responding to the *Telex* case on behalf of defendant, HYPO offers an alternative explanation of *Telex* (i.e, that it involved employee payments and use of development notes), asserts that there should be an exception to the rule of *Telex* because Version 1 involved disclosures, and poses a hypothetical counter-example consisting of the facts of the *Telex* case but where there were numerous voluntary disclosures of

[12] The former is preferred because a response that distinguishes by finding an alternative explanation rejects the rule of the cited case; the response that distinguishes by finding an exception accepts the rule of the cited case as applied to the fact situation.

[13] *Telex* is relevant because the *competitive-advantage-gained* dimension also applies to fact situation 1.

confidential information. Although the choice is close, the response to the *Telex* case is ranked as weaker because the counter-example was a hypothetical rather than a real case. In selecting plaintiff's best case, the program leads off with the *Telex* case.

6.6 The EXPLANATION Module

In HYPO, the EXPLANATION module explains:

1. a request for additional facts of the FACT-GATHERING module;
2. a point and response exchange generated by the ARGUMENT module; and
3. what claim a party should make and the strongest points for and against the claim.

HYPO explains a factual query to the user by posing a case example, real or hypothetical, that illustrates the reason for the request. The examples are derived from the cases indexed by the near-miss dimension that motivates the question.

In the fact-gathering example of Section 6.4, if the user demands to know why the program asked whether Widget-King had made any disclosures, the program responds, for example, that it is trying to build an argument for the defendant, that in the *Midland-Ross* case, where plaintiff had disclosed the trade secrets to 100 people, defendant won. In giving the explanation, the program summarizes only the parts of the example case that are relevant to illustrating the significance of the solicited fact by referring to the prerequisites and focal slots of the near-miss dimension that motivated the question.

Since explanations of factual queries are made with examples, the query need not be expressed as a question. Instead of asking whether the trade secret information was disclosed to anyone and then waiting for the user to ask why, the COUNSELOR system could pose a hypothetical: "**What if** Widget-King disclosed the confidential information to 100 people? In the *Midland-Ross* case, the court held for the defendant where...." In so posing the query, the system takes the initiative, retains control over the question-asking process and asks a more meaningful, pointed question initially. The EXPLANATION module calls upon the HYPO-GENERATOR to build this hypothetical variant of Version 3. See Section 6.7.

In explaining a point/response exchange, the program has to:

1. compare the current fact situation and a cited case to focus on the relevant similarities when the case is used to make a point or on the significant differences when the case is being distinguished.
2. describe how a real counter-example is covered by the rule of a cited case but reached the opposite result;

3. describe the relevant features of a hypothetical counter-example to which the rule of a cited case applies and emphasize the facts that should lead to an opposite result.

The selection of what facts are relevant is made by referring to the prerequisites and focal slots of dimensions that have been used in the ARGUMENT module to generate the point or response.

The program can illustrate the significance of certain facts for the strength of a point by hypothetically changing them. For example, the program can create a hypothetical based on a fact situation but where a focal slot value has been shifted to weaken the point or a dimensional prerequisite has been removed. See Section 6.7. By generating a new response to the point, the program can illustrate the effect the changed fact has on the opponent's ability to respond to the point.

To explain what claim a party should make and the strongest points pro and con, the program chooses the claim asserted in the cases that the BEST CASE SELECTOR has chosen. It supports the claim by presenting and explaining points based on those cases. The EXPLANATION module plays Devil's Advocate, too. It presents the responses to the party's best cases and cites the opponent's best cases. The module also explains why not to cite a particular case by demonstrating what the opponent's strong response would be.

6.7 The HYPO-GENERATOR Module

Basically the HYPO-GENERATOR starts with a given fact situation, or *seed* case, and generates legally relevant or plausible derivative hypotheticals which are modifications of the seed case. Since one cannot explore all the "legally" possible hypos (in the sense of syntactic legal move), one needs to explore the space heuristically. The dimensions provide a handle on how to do this exploration in a legally meaningful way.

The process occurs in two steps:

1. analyze the seed case;
2. generate legally relevant derivative hypotheticals.

Step one is accomplished by the CASE-ANALYSIS module and results in the case-analysis-record described in Section 6.2.

Step two is accomplished by the HYPO-GENERATOR which, given a specification from the ARGUMENT, FACT-GATHERING or EXPLANATION modules, uses the case-analysis-record,

and heuristics like the following to generate hypotheticals derived from the seed case:

H.1 Pick a near miss dimension and modify the facts to make it applicable.

H.2 Pick an applicable dimension and make the case weaker or stronger along that dimension.

H.3 Pick a dimension related to one of the applicable dimensions and apply 1 or 2.

H.4 Pick an applicable dimension and make the case extreme with respect to that dimension.

H.5 Pick a seed case that is a win and, using 1 and 2, move it toward a target case to create a near win.

With a specification provided by the EXPLANATION and FACT-GATHERING modules and its own heuristics for building hypos, the HYPO-GENERATOR constructs a hypothetical that is not only legally sensible but tailored to the needs of explaining a particular question or point. The specifications include the seed and target cases and the dimensions along which the seed is to be stronger or weaker.

The *Telex*-based hypothetical of Section 6.3.2 is an example of a near win hypo (H.5) in which the seed case, *Telex*, is "moved" in the direction of the target case, Version 2 of the Sample Fact Situation, using H.1 and H.2, with the result that an argument can be made that the hypo should be decided for the plaintiff as in the *Telex* case.

The hypothetical variant of Version 3, described in Section 6.4, was generated using a specification in which the FACT-GATHERING module tells the HYPO-GENERATOR to build a hypo based on Version 3 as seed to which the near-miss dimension *secrets-voluntarily-disclosed* applies and to make the hypo as weak for the plaintiff along that dimension as the target case, *Midland-Ross*. The HYPO-GENERATOR makes the near-miss dimension apply to the hypo (H.1) by adding facts to the seed case corresponding to the dimension's missing prerequisite (i.e., 100 disclosures, as in *Midland-Ross*.) The hypo can be made stronger or weaker for the plaintiff (H.2) by modifying the facts of the seed case corresponding to the focal slot of the dimension (e.g., increasing the number of disclosees.) The dimension's focal-slot, range and direction-to-strengthen-plaintiff slots contain the information necessary for making the hypo stronger or weaker along the dimension.

HYPO's facility for generating hypotheticals allows the program to guide the user through the space of possibly significant facts and precedent cases and to explain their significance. For a more complete description of the HYPO-GENERATOR, see [Rissland & Ashley, 1986].

6.8 HYPO's Architecture

HYPO's control structure is designed to demonstrate the utility of the case-based method for asking pertinent questions, making and responding to points, and generating meaningful hypothet-

icals in explaining the queries and argument exchanges. Figure 5 shows an overview of HYPO's architecture. The typical flow of control is as follows:

1. As the initial input, the user provides an incomplete fact situation and an instruction to make a point on behalf of plaintiff. INTAKE and COMMAND modules, not described here, provide a menu-driven facility for the user's specification of facts and commands.
2. The CASE-ANALYSIS module fills in the initial case-analysis-record including relevant cases from the CKB.
3. The FACT-GATHERING module poses questions about possible additional facts that would lead to other pro-plaintiff cases. It calls the EXPLANATION module to justify the questions with examples. EXPLANATION calls the HYPO-GENERATOR to make hypothetical variants of the fact situation to use in the explanations. This process continues until the near-miss dimensions have been exhausted.
4. The ARGUMENT module calls the BEST CASE SELECTOR to decide which case to cite, generates the corresponding point and calls the EXPLANATION module to explain the point by example.
5. The COMMAND module offers to respond to the point. To facilitate responding, the FACT-GATHERING module, in Devil's Advocate mode, asks questions about distinguishing facts that lead to pro-defendant cases.
6. The ARGUMENT module constructs a response. It instructs the HYPO-GENERATOR to make hypothetical exceptions to the rule of the cited case. The EXPLANATION module presents the response.
7. The COMMAND module offers to generate a counter-response or make another point.

In the course of the above flow of control, the program needs to make a number of decisions including the following (Section numbers refer to discussions of the decision making methods.):

- Whether to ask a question to supplement the fact situation and which question to ask next. (Section 6.4)
- Which is the best case a party can cite? (Section 6.5)
- Which point should the program respond to? (Section 6.5 describes methods that can be used to select the weakest point.)
- When to use a hypothetical to make a point and what kind of hypo. (Section 6.7)

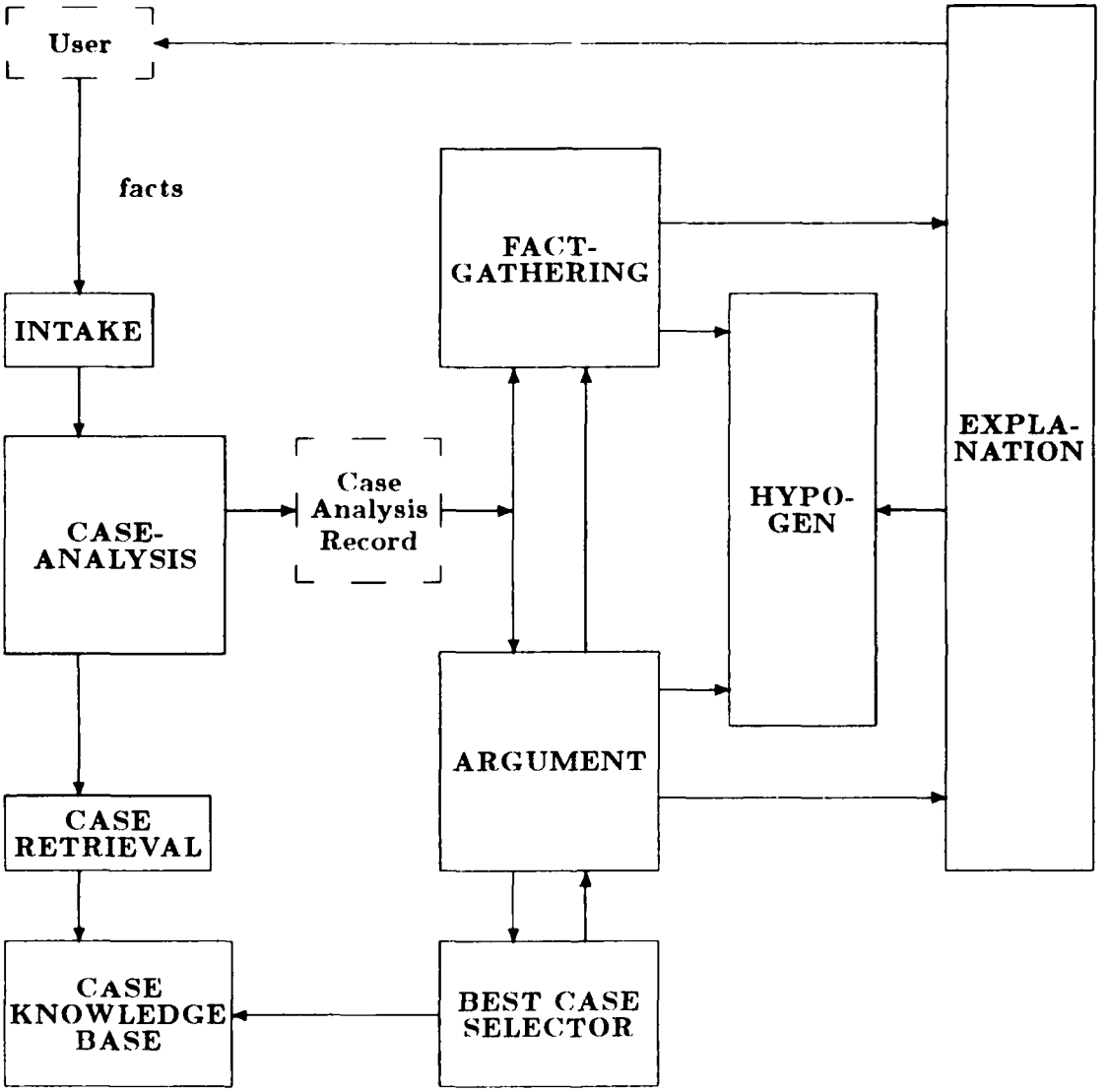


Figure 5: Overview of HYPO's Architecture.

7. Expected Results and Discussion

7.1 How the Research Should be Evaluated

The thesis research should be judged by the program's ability to make and respond to legal points about a fact situation, to ask sensible questions about missing facts, and appropriately to use novel hypotheticals in the course of its interaction with the user. The program will be implemented employing the trade secrets domain and parts of related legal domains like the laws of contracts and copyright. The dimension-based approach of the program will be applied by hand to an unrelated legal domain to establish its generality.

The program should not be regarded as a legal expert system; the program is not intended to come to the same legal conclusions about a fact situation that an attorney would. Nor will it manifest the same understanding of the implications of cases in the domain as that of an attorney. It is hoped, instead, that the program's argument moves and questions will impress an attorney as being reasonable and intelligent from a legal viewpoint. Where the program makes a point about a fact situation that, although reasonable, is wrong from a lawyer's viewpoint, it is hoped that the program can exploit the weakness when asked to respond to the point. The program will be a success if, like an attorney, it always makes the most effective response available to it under the circumstances.

In order to provide an objective standard with which to compare HYPO's performance, a number of experiments will be undertaken:

- A fact situation and summaries of cases available to HYPO will be provided to lawyers or law students who will be asked to make or respond to points on behalf of the parties in the fact situation. The same fact situation will be submitted to HYPO. The points and responses will be compared. See [Yu, et al., 1984].
- A real case will be selected. All of the cases cited in the real case will be entered into the CKB. The fact situation of the real case will be submitted to HYPO. The argument points and responses generated by HYPO will be compared to those of the opinion in the real case.
- A fact situation will be submitted to HYPO and queries based on the fact situation will be submitted to a computerized legal search program like LEXIS. The cases deemed relevant by HYPO will be compared to those returned by the database query.

7.2 Assumptions and Simplifications

Although case-based reasoning is a very important tool in legal argument and one that is applicable in many legal domains, the proposed thesis research does not deal with all kinds of legal reasoning. Many legal issues turn on the logical interpretation of the language of a statute

or contractual provision. Sometimes courts have already interpreted that or similar language in the context of a real case; then the case-based methods of this research would apply. Frequently, however, there are no cases to guide the attorney who must then rely on other kinds of reasoning to justify his interpretation.

This research assumes that the roles of legal rules and predicates are mainly to assist in finding relevant cases and in formalizing an explanation of a decision that has been made by case-based reasoning. As discussed in Section 8., other researchers put far more emphasis on using legal rules and predicates to reason about and analyze a fact situation. By contrast, the thrust of the thesis research is to show how much legal reasoning can be accomplished in a computationally inexpensive way by appropriately exploiting a database of case examples.

Given the above assumption, however, the HYPO program still does not adequately model the explanation and case-finding roles played by predicates and rules. To expand the explanation facility, it would be necessary for the program to express the significance of dimensions in more abstract terms. As previously discussed, a dimension's significance is that the facts it represents are a plausible reason for deciding a claim. The program needs to re-express that reason in more abstract terms, perhaps by relating it to rule-like definitions of the elements of a claim. This will provide yet another interpretation of a case to which the dimension applies, one that is not authoritative but which can be used as an explanation of why the case was decided as it was. In keeping with the spirit of the HYPO approach, we would also want the program to debunk such an explanation by finding alternatives and counter-examples.

Interestingly, that same abstraction of a dimension's significance may assist the program to find relevant cases and to build hypos whose facts, though not so obviously similar to the fact situation or to the seed case, raise a similar abstract issue. In effect, the computational definition of relevance in HYPO will be expanded to include not only cases to which a common dimension applies, but those to which the abstract expression of a dimension's significance applies.

Although while developing the feasibility of case-based reasoning techniques, it has been convenient to maintain a small number of dimensions and cases, the need to scale up the size of the Case Knowledge Base raises issues about automating the process of acquiring new cases and dimensions and about the effieciency of the CASE-ANALYSIS, BEST CASE SELECTION and FACT-GATHERING modules that have not been addressed here. The HYPO program currently has a dozen cases and eight implemented dimensions. We have identified some thirty relevant dimensions in the trade secrets domain, each involving two or more cases.

As part of the COUNSELOR project, a natural language generation and understanding system, the HYPO system has been designed to motivate the system to generate natural language queries, explanations by example and argument exchanges and to understand the fact patterns, questions and other inputs of the user. HYPO helps to decide what is relevant to express or hear in a particular context but does not itself express or understand natural language outputs or inputs. Nor does HYPO deal with natural language texts of cases. As indicated previously, HYPO's cases

are represented with frames that have been filled by a human who has read and interpreted the written opinion of a case.

8. Related Work

8.1 AI Models of Legal Reasoning

Previous models of legal reasoning have recognized the desirability of designing a program to reason with cases to deal with the "open-textured" meanings of legal predicates and to argue both sides of a legal issue, but no one has built a working program that does both or one that employs hypotheticals in the argument.

Gardner has developed a rule-based system which identifies legal issues in the analysis of a law school examination fact pattern involving the contracts law of offer and acceptance. The program has three heuristics for distinguishing "hard" and "easy" legal issues including one that involves trying to match a fact situation to positive and negative examples of a legal predicate. If there is no match or both match, then whether the predicate applies is treated as a hard question. The examples are abstracted from real cases, are indexed by the legal predicates of which they are positive or negative examples, and consist of simple patterns including only the facts relevant to satisfying a particular predicate. Gardner refers to, but has not implemented, a final stage of reasoning which would produce arguments on both sides of hard questions; if implemented, this stage would make use of the full representation of real cases, rather than the abstracted examples [Gardner, 1984].

In Taxman II, McCarty presents a knowledge representation scheme for representing legal concepts involved in determining whether a particular kind of corporate distribution is taxable. A concept is represented by a three part data structure: a logical template that specifies the necessary but not sufficient conditions for the concept to apply, a set of exemplars or cases, real or hypothetical, where the concept does or does not apply, and a set of transformations that specify how one gets from one exemplar to another. The transformations specify what features of two situations can be compared (e.g., the before and after ratios of stock owned by distributees.) The papers also present a theory of legal argument in which the participants adopt moves: A proponent asserts that his client should win because his fact situation is an example of a particular concept. He attempts to map the facts of his situation into an example where a similarly-situated party won. His opponent poses a counter example that also fits the concept but where the opposite result is called for [McCarty & Sridharan, 1982]. [14]

Although the Taxman II project presents a knowledge representation scheme for case-based legal argument, it does not appear that the program generates a legal argument or that more than

[14]McCarty is also working on a theory of permissions and obligations to represent the legal effects of actions [McCarty, 1985]. The relation of this work to the Taxman II project is not clear.

one case has been represented in detail using the scheme. McCarty does not specify what task the program accomplishes or what control structure is employed [McCarty & Sridahran, 1981; 1982].

Waterman has developed a rule based expert system to model how lawyers estimate the value of products liability and negligence cases. It is intended to be part of a larger system to model tactical decision making in negotiating a settlement. The research does not deal with legal argument. Waterman estimates that it would take thousands of rules to model estimating the value of auto accident cases raising products liability issues. Although the authors recognize the problem of using rules to represent the meanings of ill-defined predicates like "unreasonably dangerous" or the "foreseeability of injury", they suggest using ever more refined rules to show how the terms were applied in particular contexts. The authors suggest, but have not implemented, comparing the facts of the hypo to prior cases in which the term was applied and providing some numerical measure of fit. In eliciting rules from experts and to debug those rules, the authors used hypotheticals suggesting that hypos are the interesting information that the system should keep around. Keeping the cases around might help the system to deal with the fact that arguably applicable legal rules may be inconsistent and to explain by example why a conclusion was not reached [Waterman & Peterson, 1981].[15]

In an early work, Meldman developed a rule-based system to analyze fact situations involving intentional tort claims like assault and battery. The program had general rules defining the elements of the claims and more specific rules abstracted from the facts and holdings of real cases involving the claims. In effect, the latter rules were generalized examples of when the individual elements of a claim were or were not satisfied. The actual facts of the cases were not used in the reasoning process. The goal of the program was to analyze a fact situation by mapping its facts to the more generalized fact situations stated in the rules. The subgoal of mapping the inputted facts to the elements of the claim is accomplished by matching the facts to those of the more specific, generalized examples. A match entails finding common facts or common ancestors in a generalization hierarchy. Unlike the HYPO research, Meldman does not attempt to model arguing the analysis both ways or distinguishing cases, and treats each case as having a unique holding [Meldman, 1977].

8.2 Argumentation

The ability to generate hypotheticals and explain with examples could be a boon to discourse modelling. The Justices use hypothetical examples probably because the hypos are compact, unambiguous mechanisms for making abstract points without dealing with abstract language. Using a hypothetical, the justices can maintain control of the argument, systematically probe the meanings of new concepts as concretely applied, and flexibly adapt to responses without becoming embroiled in disputes about abstract definitions of legal concepts. The possibility of manipulating fact situations to illustrate abstract concepts without having to control inferencing about abstract terms

[15] Other proponents of rule based expert systems in the legal domain include Michaelsen,1984. See also Sergot, et al., 1986].

could make generating sophisticated discourse about cases a lot easier.

The ARGUMENT, BEST CASE SELECTION and EXPLANATION modules described above provide a capability for case-based inferencing with which it will be possible to implement a working discourse program that can discuss legal cases. As components of a Higher Level Argument Module these modules provide the program a way to choose which cases to cite in support, which to cite against the opponent, which cases it can distinguish and how to distinguish them. The Higher Level Argument Module will need a way of knowing how much support it can muster for the various claims and defenses it can assert, a control mechanism to plan the program's construction of an argument and counter-arguments and a way to model topic transitions in an argument. Research on discourse, argument and rhetoric reported in [Flowers, et al., 1982; Birnbaum et al., 1980; McGuire, et al., 1981; Birnbaum, 1982; Cohen, R., 1983; McDonald & Pustejovsky, 1985; Perelman, 1982; and Reichman-Adar, 1981; 1984] is relevant in this connection.

The Higher Level Argument Module will also need tactical knowledge to carry out such argument strategies identified by Rissland as distinguishing or attacking a rule with counter-examples such as: how to "bolster" a case, when to avoid raising an issue so as to "minimize its exposure", how to "moot" an opponent's point, how to "focus" or "shift" the focus of an argument, [16]and how to obfuscate an issue by throwing out red herrings, which order to make one's points in, when an issue should be conceded, when one should raise a weakness in one's case oneself rather than wait for the opponent to do so, etc. [Rissland, 1985].

We also want the Higher Level Argument Module to employ other high level argument strategies as "slippery slope" sequences [Rissland, 1984a] to attack rules. This particular strategy involves attacking a proposed rule by creating a sequence of hypos such that it leads to a case where the rule implies an unjust result. One then either argues that the current case is like the "reductio" hypo of the sequence or that the rule is unsound.

8.3 Memory Organization and Case-Based Reasoning

Recent research on memory organization bears on how HYPO's dimension-based indexing scheme can be supplemented automatically to introduce new dimensions and new relations between dimensions.

Kolodner has developed a question-answering system called CYRUS with a database that reorganizes its indexing scheme and representations of events as new information is added [Kolodner, 1983a; 1983b]. The system indexes events according to the aspects of an event that differ from the norms of the conceptual category of the event - whether they violate expectations, are more specific, more general, or incorporate additional features. When CYRUS encounters a new event that is similar to an event already in the database, it generalizes them by creating an E-mop with

[16]Note that responding to a case by providing an alternative explanation or by posing a counter-example shifts the focus of an argument away from the details of the fact situation to a discussion of the cited case or rule.

their similarities as the E-mop's norms and with each event indexed under the E-mop according to its peculiarities. Statistical methods are used for making and testing the usefulness of the generalizations employed as indexes.

Building on this memory representation scheme, Simpson and Kolodner have developed a case-based reasoning program that solves problems in the domain of mediating disputes [Kolodner, Simpson & Sycara-Cyranski, 1985]. The MEDIATOR program has a knowledge base of case events dealing with physical, economic, and political disputes and representations for common mediation failures, tactical plans for solving disputes, and ways to correct failed plans. Cases are indexed by their features, including the disputants, their goals, disputed objects, and any features which caused failures in mediating the dispute. For example, an attempt to mediate a physical dispute between two sisters over an orange by dividing it equally, which failed because one sister wanted the rind and another wanted the fruit, could be recalled by any physical dispute where the tactic of "divide equally" failed or where the object of dispute is something edible. The orange dispute was solved by dividing the object into different parts, skin and fruit. When recalled in connection with a physical dispute where "divide equally" fails, the orange dispute leads the program to try the alternative solution plan. In MEDIATOR, as distinguished from HYPO, the evaluation function for selecting a best case from many that are retrieved by the reminding process takes into account only the closeness of fit to selected features of the current case. A priority is associated *a priori* with the different feature types so that the retrieved cases can be ranked.

8.4 Explanation

Clancey has raised a number of issues about explanation in expert systems and tutoring that bear on this research. For Clancey, explanation is a matter of relating features of a fact situation to a model of the domain. According to Clancey,

> [p]rinciples of a [causal domain] model are good for summarizing arguments, and good to fall back on when you've lost grasp on the problem, but they don't *drive the process* of medical reasoning [Clancey, 1983, p. 233].

If rules defining the elements of legal claims are seen as a domain model, the statement is entirely translatable to the legal domain. Such rules, do not drive legal reasoning, as Gardner, Meldman, and perhaps Waterman would have it; the rules help to organize and explain points in an argument as well as to find relevant cases.

Clancey's principle mechanisms for explanation are to cite a more abstract rule that relates the rule being explained to the concepts of a causal model of the domain or to a strategic model of the task of diagnosis [Clancey, 1983, p. 248]. But he has missed the utility of citing a well-engineered hypothetical or real example to illustrate and explain a point. He recognizes that diagnostic rules are arguments that a process has occurred in a particular way [Clancey, 1983, p. 230]. But he does

not mention that a diagnosis can be argued both ways and that good tutors invite their students to do just that.[17]

8.5 Reasoning with Examples

Rissland has emphasized the importance of hypotheticals in the development and "debugging" of arguments in law [Rissland, 1983; 1985], mathematics and computer science [Rissland, Valcarce & Ashley, 1984; Rissland, 1984b] and of the dialectic in which generating rules drives the search for examples and examples provide feedback for the revision of rules [Rissland, 1984b]. This research builds on her experience with example generation and example-based reasoning [Rissland & Soloway, 1980; Rissland, 1981; Wall & Rissland, 1982].

8.6 Analogical Reasoning

In the HYPO program, although the task of matching fact situations to cases involves judging the similarity of features, what counts as similar is determined dynamically by the use to be made of the cases in an argument. Matching in HYPO is accomplished in three stages by the CASE-ANALYSIS, FACT-GATHERING and BEST CASE SELECTION modules. In CASE-ANALYSIS and FACT-GATHERING, the definition of the dimensions insures that the program tries to match only features relevant to deciding a claim. The goals of building a case or playing Devil's Advocate focusses FACT-GATHERING on those relevant cases most useful for supporting or attacking a claim about the fact situation. In BEST CASE SELECTION, the utility of a case for making a point to which there is no strong response further refines the determination of similarity. The comparison of the strengths of responses is the symbolic measure of how close a functional analogy there is between a case and a fact situation.

In HYPO, whether a case matches a fact situation depends on its utility and that depends dynamically on the context in and purpose for which the case is to be used as well as the state of the cases in the CKB indexed by the applicable and near-miss dimensions. Research on analogical reasoning illustrates some of these distinctions. See [Ashley, 1985].

Winston developed a program that reasons about legal fact hypotheticals like simple assault cases. The program uses a matching process to decide if a base situation is analogous to a target situation. The program attempts to place the parts of the target situation into correspondence

[17]Cohen has criticized the fact that rule-based expert systems rely on numerical probabilities or "certainty factors" to deal with uncertainty, for example, in medical diagnoses. He has attempted to provide a program with symbolic, as opposed to procedural or numerical, means for managing uncertainty, in part so that a program can better explain its reasons for making certain decisions. HYPO's BEST CASE SELECTION and EXPLANATION modules, with their emphasis on selecting the best case by arguing both sides of a point, can be viewed as providing another, case-based approach to reasoning about and explaining a decision for which both sides have support. The criteria for comparing responses to points play the role of endorsements in Cohen's research [Cohen, P., 1984].

with the base situation by matching up the objects, their classes and properties and by matching up the acts and relations that link objects. Points are scored for successful matches. The match with the highest total points is deemed to indicate an analogous base situation. [Winston, 1980, pp. 693–697]. In order to reduce the number of possible matches that need to be considered, some relations, those that involve causation and temporal relations, are considered to be more important. For the purpose of matching, other relations among the objects are ignored [Winston, 1980, pp. 695–696].

Burstein's work suggests a way to improve on matching. Burstein's CARL employs abstract structures to selectively and incrementally map features between the target and base domains. The mapping is performed whenever the structure appears to allow the program to make useful inferences. Only the information that will be useful for making the inference is mapped. By contrast, matching tries to link up information in the base and target domains whether there is reason to believe the linkages will generate useful inferences or not [Burstein, 1983].

Both Winston and Burstein have demonstrated how analogies provide roles for facts to play in an interpretation of a case. By identifying the base situation as analogous to the target situation, even though one may not fully understand the meanings of facts in the target situation, one may still know the roles that corresponding facts played in an interpretation of the base situation and be able to hypothesize that the target's facts play the same role.

Carbonell's programs illustrate the need to find ways of recognizing analogies that overcome the problem that situations are represented in different ways or at different levels of abstraction [Carbonell, 1983].

Kedar-Cabelli has emphasized how the purpose for which an analogy is to be made can be used to select among possible analogies. She suggests, for example, that the purpose of a statute can be used to select the kind of analogy to draw to show that the statute has or has not been violated [Kedar-Cabelli, 1984]. Unfortunately the purposes of statutes are notoriously difficult to ascertain. As this proposal makes clear, however, the context of a legal argument does provide a purpose that can be used to guide selection of analogous cases or hypotheticals, namely the purpose to support or attack a party's claim.

8.7 Learning

The HYPO research bears on a number of issues in machine learning.

In its generation and use of conflict examples, HYPO can be compared to Lenat's discovery learning approach in AM. HYPO generates some of the conflict examples hypothetically and uses them to uncover the limits of the significance and range of a dimension. Conflict examples show where the interpretation of a case according to one dimension conflicts with interpretations according to other applicable dimensions. In AM, special examples like the empty set, zero, one and boundary examples of a concept are used to determine the relations between discovered concepts,

whether one is a specialization or generalization of the other, and to discover new concepts. Concepts are adjudged to be interesting, in part, because of features of the concept's examples, how many examples there are, how easy they are to generate, whether they are all examples of other concepts like perfect squares, etc. [Lenat, 1977]. In HYPO, the program discovers novel hypothetical fact situations which it recognizes as interesting because of the utility of the hypothetical in making a response to a point in a legal argument. Unlike AM, HYPO does have a performance task that can be improved by the discovery of the new items, namely FACT-GATHERING. Unlike AM, HYPO does not discover new concepts; HYPO does not, for example, learn new dimensions.

Learning new dimensions could mean any or all of the following:

1. Understanding natural language opinions of cases well enough to determine that a court regarded a particular collection of facts as a reason for deciding the case in a particular way;
2. Generalizing from the non-natural language representations of cases in the database that certain collections of facts lead to deciding a certain claim for plaintiff (or defendant). The facts are assumed to be those for which primitives have been defined in the representation scheme or which can be inferred from primitives.
3. Generalizing as in the previous item but where new terms are introduced into the representation scheme for representing new facts or collections of facts.

Machine learning research on inducing rules from instances bears on the latter two items. Since cases represent noisy data for the existence of rules for how to decide claims and since rules can be expected to need modification in the light of new cases, a version space approach like that of Buchanan and Mitchell may be be appropriate [Buchanan & Mitchell, 1978; Mitchell, Utgoff & Banerji, 1983]. It should be noted that the version space approaches fashion rules consistent with all the training instances. A dimension indexes examples that are both consistent and inconsistent with its significance because, in the legal domain, as in most domains, there are multiple, sometimes inconsistent, interpretations of a single event. This feature complicates the problem of appropriately generalizing from negative instances.[18] See also [Rissland, et al., 1986].

Research on the "New Term" problem by Utgoff and Rendell bears on the third sense of learning a new dimension. Utgoff's method for learning new terms is not really inductive but deductive; it involves propagating back constraints satisfied by some positive instance [Utgoff, 1983; Mitchell, et al., 1985]. To employ Utgoff's method, one would have to define explicitly the criteria for winning a particular claim.

[18]HYPO needs to be concerned with local consistency among cases (e.g., it should not cite a case on behalf of a party if the case can be construed as contradicting some other point of the party.) That is to say local consistency with what has been said should be factored into the selection of the best case and the program should be prepared in response to point out a local inconsistency. But the fact that a rule the program induces from a case and cites on behalf of a party may be globally inconsistent, (i.e., may be contradicted by some other case in the database) is the stuff of which responses are made and why selecting the best case is important and hard.

Rendell's approach is inductive; the program employs utility for winning a game of checkers to compare board positions abstractly and to select salient, generalizable features of board positions that may be useful to include in an evaluation function [Rendell, 1985]. In the HYPO context, utility of a set of features in making a point in an argument might serve the same function.

This raises the other learning issue implicit in HYPO: learning the evaluation function for selecting the best case by comparing the strengths of responses. The evaluation function is used to search, using a minimax procedure, a game tree representing the points that a party could make for the point that it should make. Note that the evaluation function depends on the state of the CKB in a particular area of the space of cases. A response to a particular case in a given fact situation may be radically changed by the addition of new cases to the CKB. A generalization approach like that in [Samuel, 1963] might involve identifying features of fact situations and of CKB configurations that are predictors of the strength of a point based on a case from within the configuration.

9. Conclusion

In this thesis proposal, we have described a computational mechanism for case-based reasoning in which data trigger reasons, reasons trigger cases, cases trigger hypotheses, and hypotheses trigger and are resolved by arguments. The data are a fact situation, the reasons are dimensions, the cases are indexed by dimensions, the hypotheses are statements that a claim applies to the fact situation or that a party in the fact situation should win or lose on a particular claim, and the arguments are the points and responses considered in selecting the parties' best cases.

Our case-based model of legal reasoning recognizes that cases are not "pure" examples of a successful or unsuccessful claim. In most legal cases, some facts favor the winner of the claim and some favor the loser. When a court decides a case in favor of one party, it implicitly or explicitly assigns a relative importance to the conflicting features of the case. The case can be used in subsequent arguments in support of assigning the same relative importance to similar features. A case, however, rarely stands for a unique assignment of relative importances to features; it is subject to a number of interpretations in light of all of its features. This is why a single dimension indexes cases that are both consistent and inconsistent with its significance, why examples of conflict among dimensions are important, and why deciding what claim succesfully applies in favor of a party in a fact situation is not simply a matter of determining what elements of a claim are satisfied, but of comparing the strengths of responses to points made in favor and against the party's position. The strength of the argument in favor of a party's claim does not depend on the meaning of a claim's elements, but on which facts are shared among the current fact situation, the cited case and other cases, including contrary ones, in that region of the Case Knowledge Base.

We have illustrated with examples of oral arguments before the Supreme Court how case examples and hypotheticals are primary tools for making arguments and have demonstrated how a

program can generate hypotheticals and cite relevant examples to make or respond to points about the legal analysis of the fact situation.

We have contrasted our model of legal reasoning with that of other AI researchers and shown how this research draws on and contributes to recent work in argumentation, case-based reasoning, dynamic memory organization, explanation and analogical reasoning.

References

[1] Kevin D. Ashley. Reasoning by Analogy: A Survey of Selected A.I. Research with Implications for Legal Expert Systems. In Charles Walter, editor, *Computing Power and Legal Reasoning*, West Publishing Co., St. Paul, MN, 1985.

[2] Kevin D. Ashley and Edwina L. Rissland. Toward Modelling Legal Argument. In Antonio A. Martino and Fiorenza Socci Natali, editors, *Atti preliminari del II Convegno internazionale di studi su Logica Informatica Diritto*, pages 97–108, Consiglio Nazionale delle Ricerche, Istituto per la documentazione giuridica, Florence, Italy, September 1985.

[3] L. Birnbaum. Argument Molecules: A Functional Representation of Argument Structure. In *Proceedings of the Second National Conference on Artificial Intelligence*, American Association for Artificial Intelligence, Pittsburgh, PA, August 1982.

[4] L. Birnbaum, M. Flowers, and R. McGuire. Towards an AI Model of Argumentation. In *Proceedings of the First National Conference on Artificial Intelligence*, pages 313–315, American Association for Artificial Intelligence, August 1980. Stanford, CA.

[5] B.G. Buchanan and T.M. Mitchell. Model-Directed Learning of Production Rules. In Waterman and Hayes-Roth, editors, *Pattern-Directed Inference Systems*, Academic Press, 1978.

[6] M. H. Burstein. A Model of Learning by Incremental Analogical Reasoning and Debugging. In *Proceedings of the Third National Conference on Artificial Intelligence*, American Association for Artificial Intelligence, Washington, D.C., August 1983.

[7] J. G. Carbonell. Learning by Analogy: Formulating and Generalizing Plans from Past Experience. In Michalski, J.G. Carbonell, and T. Mitchell, editors, *Machine Learning: An Artificial Intelligence Approach*, Tioga Publishing, CA, 1983.

[8] W.J. Clancey. The Epistemology of a Rule-based Expert System: A Framework for Explanation. *Artificial Intelligence*, 20(3):215–251, 1983.

[9] Paul R. Cohen. *Progress Report on the Theory of Endorsements: A Heuristic Approach to Reasoning about Uncertainty.* Technical Report 84-15, Department of Computer and Information Science, University of Massachusett, Amherst, MA, 1984.

[10] Robin Cohen. *Computational Model for the Analysis of Arguments.* Technical Report CSRG-151, Department of Computer Science, University of Toronto, 1983.

[11] M. Flowers, R. McGuire, and L. Birnbaum. Adversary Arguments and the Logic of Personal Attacks. In W. Lehnert and M. Ringle, editors, *Strategies for Natural Language Processing*, Lawrence Erlbaum Associates, Hillsdale, NJ, 1982.

[12] Li-Min Fu and Bruce G. Buchanan. *Inductive Knowledge Acquisition for Rule-Based Expert Systems.* Report KSL-85-42, Knowledge Systems Laboratory, Department of Computer Science, Stanford University, 1985. To Appear in *AI Journal.*

[13] A. vdL. Gardner. *An Artificial Intelligence Approach to Legal Reasoning.* PhD thesis, Department of Computer Science, Stanford University, 1984.

[14] M. R. Gilburne and R. L. Johnston. Trade Secret Protection for Software Generally and in the Mass Market. *Computer/Law Journal*, III(3), 1982.

[15] S. Kedar-Cabelli. *Analogy with Purpose in Legal Reasoning from Precedents A Dissertation Proposal.* Technical Report LRP-TR-17, Laboratory for Computer Science Research, Rutgers University, 1984.

[16] Janet L. Kolodner. Maintaining Organization in a Dynamic Long-Term Memory. *Cognitive Science*, 7(4):243–280, 1983.

[17] Janet L. Kolodner. Reconstructive Memory: A Computer Model. *Cognitive Science*, 7(4):281–328, 1983.

[18] Janet L. Kolodner, Robert L. Simpson, and Katia Sycara-Cyranski. A Process Model of Case-Based Reasoning in Problem Solving. In *Proceedings of the Ninth International Joint Conference on Artificial Intelligence*, International Joint Conferences on Artificial Intelligence, Inc., Los Angeles, CA, August 1985.

[19] D. B. Lenat. Automated Theory Formation in Mathematics. In *Proceedings of the Fifth International Joint Conference on Artificial Intelligence*, pages 833–841, International Joint Conferences on Artificial Intelligence, Inc., Cambridge, MA, August 1977.

[20] L. Thorne McCarty. Permissions and Obligations. In Charles Walter, editor, *Computing Power and Legal Reasoning*, West Publishing Co., St. Paul, MN, 1985.

[21] L. Thorne McCarty and N.S. Sridharan. *A Computational Theory of Legal Argument*. Technical Report LRP-TR-13, Laboratory for Computer Science Research, Rutgers University, 1982.

[22] L. Thorne McCarty and N.S. Sridharan. The Representation of an Evolving System of Legal Concepts: II. Prototypes and Deformations. In *Proceedings of the Seventh International Joint Conference on Artificial Intelligence*, International Joint Conferences on Artificial Intelligence, Inc., Vancouver, B.C., August 1981.

[23] D. D. McDonald and J. Pustejovsky. Multi-level, Description-Directed, Goal-Driven Natural Language Generation. In *Proceedings of the Ninth International Joint Conference on Artificial Intelligence*, International Joint Conferences on Artificial Intelligence, Inc., Los Angeles, CA, August 1985.

[24] R. McGuire, L. Birnbaum, and M. Flowers. Opportunistic Processing in Arguments. In *Proceedings of the Seventh International Joint Conference on Artificial Intelligence*, pages 58–60, International Joint Conferences on Artificial Intelligence, Inc., Vancouver, B.C., August 1981.

[25] Jeffrey A. Meldman. A Structural Model for Computer-Aided Legal Analysis. *Journal of Computers and the Law*, 6:27–71, 1977.

[26] R. H. Michaelsen. An Expert System for Federal Tax Planning. *Expert Systems*, 1(2), October 1984.

[27] T.M. Mitchell, P.E. Utgoff, and R.B. Banerji. Learning by Experimentation: Acquiring and Refining Problem-Solving Heuristics. In Michalski, Carbonell, and Mitchell, editors, *Machine Learning An Artificial Intelligence Approach*, pages 163–190, Tioga, 1983.

[28] Mitchell, Tom M. and Mahadevan, Sridhar and Steinberg, Louis I. LEAP: A Learning Apprentice for VLSI Design. In *Proceedings of the Ninth International Joint Conference on Artificial Intelligence*, pages 573–580, International Joint Conferences on Artificial Intelligence, Inc., Los Angeles, CA, August 1985.

[29] C. Perelman. *The Realm of Rhetoric*. University of Notre Dame Press, 1982.

[30] R. Reichman. Modeling Informal Debates. In *Proceedings of the Seventh International Joint Conference on Artificial Intelligence*, International Joint Conferences on Artificial Intelligence, Inc., Vancouver, B.C., August 1981.

[31] R. Reichman-Adar. Extended Person-Machine Interface. *Artificial Intelligence*, 22:157–218, 1984.

[32] L. Rendell. Substantial Constructive Induction Using Layered Information Compression: Tractable Feature Formation in Search. In *Proceedings of the Ninth International Joint Conference on Artificial Intelligence*, pages 650–658, International Joint Conferences on Artificial Intelligence, Inc., Los Angeles, CA, August 1985.

[33] Edwina L. Rissland. Argument Moves and Hypotheticals. In Charles Walter, editor, *Computing Power and Legal Reasoning*, West Publishing Co., St. Paul, MN, 1985.

[34] Edwina L. Rissland. *Example Generation*. Technical Report 81-24, Computer and Information Science Department, University of Massachusetts, Amherst, MA, 1981.

[35] Edwina L. Rissland. Examples in Legal Reasoning: Legal Hypotheticals. In *Proceedings of the Eighth International Joint Conference on Artificial Intelligence*, International Joint Conferences on Artificial Intelligence, Inc., Karlsruhe, Germany, August 1983.

[36] Edwina L. Rissland. Hypothetically Speaking: Experience and Reasoning in the Law. In *Proceedings First Annual Conference on Theoretical Issues in Conceptual Information Processing*, Georgia Institute of Technology, Atlanta, GA, March 1984.

[37] Edwina L. Rissland. The Ubiquitous Dialectic. In *Proceedings Sixth European Conference on Artificial Intelligence*, Elsevier (North-Holland), Pisa, Italy, September 1984.

[38] Edwina L. Rissland and Kevin D Ashley. Hypotheticals as Heuristic Device. In *Proceedings of the Fifth National Conference on Artificial Intelligence*, American Association for Artificial Intelligence, August 1986. Philadelphia, PA.

[39] Edwina L. Rissland, Bruce G. Buchanan, Paul Rosenbloom, and H. T. Ng. *Intelligent Example Selection: Empirical Experiments with Near-Misses*. 1986. Submitted for Publication.

[40] Edwina L. Rissland and E. M. Soloway. Overview of an Example Generation System. In *Proceedings of the First National Conference on Artificial Intelligence*, American Association for Artificial Intelligence, Stanford, CA, August 1980.

[41] Edwina L. Rissland, E. M. Valcarce, and Kevin D. Ashley. Explaining and Arguing with Examples. In *Proceedings of the Fourth National Conference on Artificial Intelligence*, American Association for Artificial Intelligence, Austin, TX, August 1984.

[42] A. Samuel. Some Studies in Machine Learning Using the Game of Checkers. In Feigenbaum and Feldman, editors, *Computers and Thought*, pages 71-105, McGraw-Hill, 1963.

[43] Sergot, M. J. and Sadri, F. and Kowalski, R. A. and Kriwaczek, F. and Hammond, P. and Cory, H. T. The British Nationality Act as a Logic Program. *Communications of the ACM*, 29(5):370-386, May 1986.

[44] SUP The Complete Oral Arguments of the Supreme Court of the United States. University Publications of America, Frederick, MD.

[45] Paul E. Utgoff. Adjusting Bias in Concept Learning. In *Proceedings of the International Machine Learning Workshop*, pages 105-109, Monticello, IL, 1983

[46] R. S. Wall and E. L. Rissland. Scenarios as an Aid to Planning. In *Proceedings of the Second National Conference on Artificial Intelligence*, American Association for Artificial Intelligence, Pittsburgh, PA, August 1982.

[47] D. A. Waterman and M. Peterson. *Models of Legal Decisionmaking*. Technical Report R-2717-1CJ, The Rand Corporation, Santa Monica, CA, 1981.

[48] Patrick H. Winston. Learning and Reasoning by Analogy. *Communications of the ACM*, 23(12):689–703, 1980.

[49] Yu, Victor L. and Fagan, Lawrence M. and Bennett, Sharon W. and Clancey, William J. and Scott, A. Carlisle and Hannigan, John F. and Blum, Robert L. and Buchanan, Bruce G. and Cohen, Stanley N. An Evaluation of MYCIN's Advice. In Buchanan, Bruce G. and Shortliffe, Edward H., editor, *Rule-Based Expert Systems*, chapter 31, pages 589–596, Addison-Wesley, Reading, MA, 1984.

END

2-87

DTIC