LESTER: Using Paradigm Cases in a Quasi-Precedential Legal Domain

Kenneth A. Lambert
Department of Computer Science
Washington and Lee University
Lexington, Virginia 24450

and

Mark H. Grunewald
School of Law
Washington and Lee University
Lexington, Virginia 24450

1 Summary

We are developing LESTER (Legal Expert System for Termination of Employment Review), a case-based reasoning program to advise in the area of unjust discharge from employment under collective bargaining agreements. LESTER uses paradigm cases to reason in a legal domain that is not governed by a strong concept of precedent. This paper describes the domain and gives an overview of the current version of the program.

2 Background

Case-based reasoning has had particular allure in the design of expert systems in law. In part, this can be explained by the apparent similarity between the techniques used in case-based reasoning systems and the process of legal reasoning in a case-based legal domain governed by the doctrine of *stare decisis* i.e., a precedent-bound domain. Both approaches assume generally that new cases must be analyzed and decided on the basis of prior decisions in similar cases. [Ashley and Rissland 1986]

In this sense, a legal expert system in a field of case-based law aims at simulating a lawyer's use of a familiar case base to predict outcome or make arguments in new cases. The known cases, however, have more than analogic value in a reasoning system; they are the law. They can be cited and relied upon in arguing or deciding. A facility that understands and can use the cases in those ways performs some of the functions legal experts perform in those domains.

An expert system for manipulating precedent thus necessarily accepts as a working premise the centrality of precedent in the expert's activity. While both similarities and dissimilarities between a new case and the case base are accorded weight, the knowledge of the system (like the body of case law itself) would be largely limited to precedent cases and precedent cases would be presumed to reflect a substantial part of an expert's knowledge of the domain. Yet in case-based legal domains where a strong concept of precedent is absent, a somewhat different role for case-based reasoning can be posited. Our work focuses on such domains.

3 The Role of Precedent

The use of precedent varies across different legal domains. Typically, in domains in which there are a limited number of formal decisions and in which there is a well-defined hierarchy of decisionmakers, the concept of precedent-bound decisionmaking exists in a strong form. On the other hand, in legal domains in which there are large numbers of informal decisions and in which decisional hierarchy is either lacking entirely or substantially obscured by the complexity of the decisional structure, the role of precedent is weaker.

The precise nature of the differences in the role of precedent in such domains and the possible explanations for them are matters of jurisprudential controversy. Without directly addressing those issues, we seek to apply case-based reasoning techniques to a domain that has a generally recognized weak notion of precedent. Our working premises are (1) that collectively the cases in this domain, as in virtually any legal domain, are understood by the expert to represent not only precedent in some sense, but also a deep, not readily modeled, legal structure and (2) that without binding the reasoning to precedent as such, a case-based expert system should be able to offer predictive and explanatory assistance to a user.

4 A Complex Legal Domain with a Weak Concept of Precedent

Lawyers who advise and litigate in the area of employee discipline governed by the "just cause" standard of collective

bargaining agreements operate in a surprisingly complex legal environment. The operative contract language, "just cause," is open-textured in the extreme. In the context of employment discipline it takes on meaning only through the decisions of private arbitrators selected by the parties to resolve specific cases. The arbitral process does not, however, convert the vague contractual language into easily-applied rules of decision:

> Just cause is hardly an obvious concept. When applying it to specific cases, arbitrators tend to define just cause in nebulous terms or to make conclusory statements. For example, "reasonable" discipline is permissible, but "arbitrary," "excessive," or "discriminatory" discipline is not. A penalty that does not "shock the conscience" of the arbitrator is upheld, but one that is not "just" under "all the circumstances" is set aside. In fact, one arbitrator characterized the term "just cause" as "purposefully ambiguous." [Abrams and Nolan]

Yet notwithstanding this high level of abstraction, experienced lawyers counsel clients and argue to arbitrators on the basis of the body of decision that has evolved. The expert presumably applies traditional legal method to this case base to produce advice, to design strategy or to make predictions.

On the surface at least, the arbitral system resembles a pure common law regime. The basic principle -- just cause for discipline -- is explicated and elaborated through case-by-case adjudication. But on closer analysis the differences may exceed the similarities. Because the arbitrator is hired by the parties to resolve a particular dispute under a particular agreement, the conventional understanding is that the arbitrator is not bound by the decisions of other arbitrators under other agreements. Formally, the process is viewed as a purely internal dispute resolution mechanism unique to the parties who structured it:

> A proper conception of the arbitrator's function is basic. He is not a public tribunal imposed upon the parties by superior authority which the parties are obliged to accept. He has no general charter to administer justice for a community which transcends the parties. He is rather part of a system of industrial self-government created by and confined to the parties. [Shulman]

Moreover:

> The labor arbitrator's source of law is not confined to the express provisions of the contract, as the industrial common law -- the practices of industry and the shop -- is equally a part of the collective bargaining agreement although not expressed in it. The labor arbitrator is usually chosen because of the parties confidence in his knowledge of the common law of the shop and their trust in his personal judgment to bring to bear considerations which are not expressed in the contract as criteria for judgment. [U. S. Steelworkers v. Warrior & Gulf]

Despite the apparent shop-specific role of the arbitrator, lawyers who work in the area understand the body of arbitral decision to represent a loose form of precedent. No other arbitrator under any other agreement is bound to follow a prior decision, but a well-reasoned decision on similar facts may have considerable persuasive authority. Nor is there any hierarchical review of the decision; but the decision of an experienced, widely-respected arbitrator in a similar case may likewise be considered particularly persuasive.

The absence of a true decisional hierarchy and inapplicability of *stare decisis* renders the body of arbitral decisions a notably inexact case base for traditional legal reasoning. Legal advice, strategy, and prediction here thus must proceed with an acute awareness that order in the field does not follow from binding precedent but (if there is order at all) from a deeper and more complex structure. The expert thus depends upon a feel for the "just cause" standard.

5 System Design

5.1 Overview

The goal of the system is to provide, on the basis of the description of a new case, an evaluation of likely outcome with supporting explanation approximating that which would be provided by a domain expert in an advisory context. The intended user would be a legal or personnel professional or paraprofessional.

The case base consists of arbitral decisions, either hypothetical or real, that represent paradigms in the just cause discharge domain or in subparts of that domain. While the actual domain consists of thousands of cases, the case base contains a small subset of these to reflect an expert "feel" for the essence of the domain or a subpart of the domain. The case base, thus, is not limited by notions of precedent, decisional hierarchy, or any other indicia of authoritativeness. Likewise, the case classification scheme used by the system avoids for the most part reliance on doctrinal rules or concepts. In these respects, our system combines some features of both precedent-oriented, casebased systems and outcome-oriented, rule-based systems. [Rissland and Ashley 1987, Smith and Deedman 1987, Greenleaf 1987]

Cases in the system are organized for most purposes around four primary dimensions, each containing a number of subdimensions. A fifth primary dimension is used for

housekeeping purposes. The primary dimensions consist, at a high level of generality, of the features of a case most likely to influence an arbitral decision. The primary dimensions used for the currently implemented subdomain, employee misconduct, are: (1) **seriousness**, (2) **mitigation**, (3) **employee status**, and (4) **process**.

The subdimensions within each primary dimension reflect the more particular features of a case that give content to the primary dimension. As explained more fully below, this structure is used both to truncate the case analysis function where one or only a few case features would likely be determinative of outcome and to maintain a context for fuller analysis in "harder" cases. The structure also serves to capture the tendency of decisionmakers in this domain to view case outcome as the resolution of tension between a set of competing values in workplace discipline for which the primary dimensions serve as proxies. Accordingly, with a few fixed exceptions, the influence of any subdimension is limited to its effect on its associated primary dimension, while primary dimensions can significantly influence one another. There are approximately 30 subdimensions for each case in the case base.

Evaluation of likely outcome, where the system is able to make a judgment, is expressed as: (1) a "sustained" grievance, (2) a "denied" grievance, or (3) a "sustained in part" grievance (normally reinstatement without backpay—an outcome commonly perceived as a compromise in this domain). The explanation is a rudimentary natural language justification for the outcome expressed as the course "most arbitrators" would take.

5.2 An Illustration

For employee misconduct, the primary dimension "seriousness" consists of nine subdimensions:

1. Type--a characterization of the type of misconduct, e.g., swearing, hitting, etc.
2. Basis--a characterization of the basis of the act if notable, e.g., racial, sexual, etc.
3. Injury--a characterization of the injury, if any, caused by the misconduct.
4. Weapon--a characterization of the weapon used, if any.
5. Single Act/Series--a characterization of the association of the act with other acts, if any.
6. Victim Status--a characterization of the relationship between the employee and the victim of the misconduct, if any.
7. Victim Sex--a characterization of the victim's sex.
8. Worksite--a characterization of the worksite.
9. On/Off Premises--a characterization of the location within the worksite (or off it) where the misconduct occurred.

The other three primary dimensions each consist of similarly relevant subdimensions. The "mitigation" dimension, for example, includes, among others, such features as provocation and spontaneity, while the "status" dimension includes length of service, quality of service record, and prior misconduct. Finally, the "process" dimension consists of characterizations of the procedural steps followed prior to discipline.

A simple case description for entry into the system would read as follows:

> J.J., a maintenance mechanic at Wellington Industries for the past 3 years, was discharged for slugging a fellow employee in the company parking lot. The victim suffered a facial laceration requiring stitches. The victim had pulled into a parking space for which J.J. was waiting. When confronted by J.J., the victim had replied, "Get lost, buddy," whereupon J.J. hit him. J.J. had a generally good work record, but had been given a five-day disciplinary suspension six months ago for throwing a tool that broke across the floor accidentally hitting another employee.

The system does not parse this description into the case representation structure. Instead, the user enters this information in the form of responses to queries generated by the system. The user selects from a range of natural language descriptors offered for each dimension and subdimension. The suitability of this approach for case entry depends upon the richness of the case representation structure and the descriptors available for choice. This quality varies across the dimensions of the current version of the program.

When the above case is processed, the final system output would read roughly as follows:

> The likely outcome in this case is: Grievance denied.
>
> Most arbitrators would hold that to hit a fellow employee on company premises causing an injury requiring medical attention is serious misconduct, warranting discharge. No sufficient mitigating factor is present. The status of the employee is unlikely to alter the outcome. There is no indication of procedural defect in administration of the discipline.

The algorithm we use to arrive at this output is described in general terms in the next section. It is important, however, to

note that at this stage in the development of the system, the case processing element is devoted primarily to achieving a reliable evaluation of likely outcome. While the output appears to reflect discrete treatment of the four primary dimensions, this is true only for the explanation function. The outcome evaluation function is integrated across the four primary dimensions. A more sophisticated explanation function should be able to achieve an equally well integrated explanation. The system output in the present version does not include references to real cases or indicate differences or similarities between the test case and real cases. This is consistent with our view of the domain as one in which it is more important for an expert system to simulate a "feel" for the body of decision generally than to generate citations to particular cases with compare and contrast references. We nevertheless recognize that there could be value in including such a function at a later point.

6 System Implementation

6.1 Operating Features

The program is written in PC Scheme, a dialect of LISP for the IBM PC and compatibles. We are currently testing the program on an IBM Personal System/2 Model 50. The program supports case base maintenance, analysis of new cases, and report generation. The program has a high-level, menu-driven, cursor-based user interface. Novice users are able to navigate through the system by typing the arrow keys, the enter key, and the escape key. File names and case names are almost the only textual information that users are required to type.

The program consists of the following modules:

1. a file handler
2. a case editor
3. a case structure editor
4. a case analyzer

The file handler supports loading and saving case bases. The case editor permits users to enter new cases into the case base, remove them or edit them. The case structure editor allows users to change the generic structure of a case and update an existing case base to conform to any changes. Typical users of the first three modules will be knowledge engineers and domain experts.

The fourth module takes the description of a new case and consults the case base for evaluation of likely outcome. The module is menu-driven, with options for entering a case description, analyzing the case, and printing a report of the analysis. A user could be any person who has an appropriate description of a case.

We have used a frame representation language (FRL) to represent the cases.[Amsterdam, 1986, 1987] The FRL uses indexing, inheritance, and demons to optimize case base maintenance, processing time and memory use. Case base searching supports boolean operations and arbitrary LISP functions in queries.

There is one frame for each case in the casebase, and one generic frame containing default information for all cases. The frame structure for each case is quite simple. Each slot in a frame is the name of a dimension for the case. The value of a slot will be any of a range of values allowed for the dimension. A value may be a number, a symbol or a list of symbols. If no value has been entered for a dimension in the frame for a particular case, the value of that dimension will default to a value inherited from the corresponding dimension in the generic frame for all of the cases.

The system keeps several special frames for purposes of indexing. The generic case frame has an instances slot which lists all of the individual cases in the casebase. Each dimension has its own frame, which has a slot listing all of the frames which have a slot for that dimension. Finally, each dimension is classified under one of five major or primary dimensions. Each primary dimension has a frame with a slot containing a list of its subdimensions.

Demons are procedures which are attached to slots. Their actions can be triggered automatically when a value on the slot is added, removed or merely examined. Our system uses demons to update the indexing of the cases when they are added to or removed from the casebase.

Inheritance, indexing, and the use of demons are features of large, complex frame-based systems. These features are particularly useful when cases differ in structure and can be organized in a taxonomy. Our current casebase is quite flat, in the sense that all cases have the same structure. Therefore, our system does not take full advantage of the power of the frame representation language. Nevertheless, we have included these features to support more highly differentiated casebases that may be developed.

Searching the case base is accomplished by pattern matching. A simple pattern consists of a dimension/value pair. A match will occur if the value of the pair in the input case pattern is the same as the value of the corresponding pair in a case frame. "Same" here means the same number, the same symbol, or a member of the list of symbols that make up the value of the case frame slot. The name of the case frame is saved whenever a match occurs.

Once again, the searching mechanism of the frame language is much more powerful than our system presently requires. Simple patterns can be combined using the logical connectives, and, or, and not, to form more complex patterns for matching. We use one of these, the or connective, to construct patterns for matching against a range of values on a dimension. This means that a match will occur if the value in the case frame is the same as one of the disjunct values in the input pattern.

6.2 Case Analysis Algorithm

In the development of the system we have modified the case analysis algorithm a number of times. The version we are now testing operates generally as follows:

1. The case analyzer module first compares each subdimension of the test case under the first primary dimension index against all of the cases in the case base. It then returns a set of the cases that produced a match along at least one dimension. Because this step yields a set of precise matches across the first primary dimension, it is an important step in the overall analysis. However, its very precision can lead to a loss of cases that are similar in some relevant respect but produce no matches. As a practical matter, this prospect is significant only if at a later point too few more precisely matched cases are available for further analysis. Thus, in a later step of the analysis, cases that have matches within a range of values on certain subdimensions are restored if the more precisely matched cases provide an insufficient basis for outcome evaluation. Otherwise, the analysis proceeds from the precise matches.

2. Beginning with the set of cases having the greatest number of dimensions matched, a poll is taken as to outcome. If the size of the case set is sufficiently large and if the poll reflects a meaningful outcome pattern, a weighting is applied to the dimensions to refine the comparison. Otherwise, the set is enlarged in steps on successively smaller numbers of matches until the size is sufficient for further processing. If the number of cases is sufficiently large after weighting, another poll is taken. If the outcome pattern is unaffected, that outcome is held as a preliminary likely outcome. Otherwise, the pre-weighting outcome is held, but with a lower degree of certainty. If an adequate set of cases cannot be obtained, processing is terminated after a limited analysis of the remaining three primary dimensions for outcome-controlling conditions, in a sense "easy" case conditions.

3. When step two yields a preliminary likely outcome, a similar match/adjust/weight routine is run against the remaining three primary dimensions. The set of cases returned for each of those dimensions is compared for commonality. The largest set of common cases across the three primary dimensions is then polled for outcome. If the poll reflects a meaningful outcome pattern and is consistent at a sufficient level of certainty with the likely outcome held from step two, that outcome is used as the system's outcome evaluation. Otherwise, the inconsistent likely outcomes returned when the match/adjust/weight routine was run independently against the three primary dimensions are evaluated with a set of heuristics to attempt to resolve the conflict.

4. If a likely outcome can be generated, the outcome explanation function is called. It draws components from each of the four primary dimensions to provide an explanation for the evaluation. Otherwise, a limited explanation for the uncertainty in the outcome is provided.

5. For purposes of prototype refinement, the system also produces various forms of intermediate output, including case tables and matrices.

7 Discussion and Conclusion

Our cast analysis algorithm is thus used primarily to position a test case within the body of arbitral decision and to identify an outcome pattern. To accomplish the positioning by associating a range of case features within a dimension or by assigning numerical weights to dimensions necessarily introduces a level of arbitrariness and departs from traditional models of legal analysis. [Ashley and Rissland 1988]. When the precise features of a test case make it an easy case, we find there is little need to associate a range of values or to assign weights in positioning. For these less interesting cases, the algorithm avoids the use of ranges or weights. When it becomes necessary to depart from that approach, as it usually does, the algorithm uses ranges before using weights, and when it is necessary to use weights, uses them almost exclusively within primary dimensions rather than across them. Apart from this general effort to minimize the use of ranges and weights, our approach to this domain accepts them as inevitable for outcome evaluation and uses them in a fairly straightforward manner. Generally, the ranges and weights simply reflect the associations and relative significance an expert would attach to particular case features, with some adjustments we have arrived at through trial and error.

The performance of the system, tested by comparing its output with actual outcome in decided cases, has improved over the period of development. Reflecting our preference to have the system decline to make an evaluation unless it can do so at a high level of confidence, it now returns an unsatisfactorily large number of declinations to evaluate. On the basis of our work to this point, we believe that continued refinement of the algorithm and some enhancement of the representational structure and case base will improve the performance significantly. After completing the additional work we now contemplate, we intend to test the system's performance against the case evaluations of a panel of domain experts.

References

Abrams and Nolan, "Toward a Theory of 'Just Cause' in Employee Discipline Cases," 1985 Duke L.J. 594 (1985).

Amsterdam, "Building a Flexible Knowledge Representation Scheme," AI Expert, November, 1986, pp. 19-22; "Retrieving Frames from a Frame Data Base," AI Expert, December, 1986, pp.

19-21; "Solving SFRL Problems with a Representation Language," AI Expert, February, 1987, pp. 15-19.

Ashley and Rissland, "Compare and Contrast: A Test of Expertise," American Association for Artificial Intelligence, 1986.

Ashley and Rissland, "Waiting on Weighting: A Symbolic Least Commitment Approach," in Proceedings AIII-88, American Association for Artificial Intelligence, August, 1988. St. Paul.

Greenleaf, Mobray and Tyree. "Expert Systems in Law: The DataLex Project," Proceedings of the First International Conference on Artificial Intelligence and Law, 1987, pp. 9017.

Rissland and Ashley, "A Case-Based System for Trade Secrets Law," Proceedings of the First International Conference on Artificial and Law, 1987, pp. 60-66.

Shulman, Reason, Contract, and Law in Labor Relations, 68 Harv. L. Rev. 999 (1955).

Smith and Deedman, "The Applications of Expert System Technology to Case-Based Law," Proceedings of the First International Conference on Artificial Intelligence and Law, 1987, pp. 84-93.

United Steelworkers of America v. Warrior & Gulf Navigation Co., 363 U.S. 574 1960.