UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION Unclassified	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution unlimited.
2b. DECLASSIFICATION / DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S)	5. MONITORING ORGANIZATION REPORT NUMBER(S) ARO 21697.11-MA
6a. NAME OF PERFORMING ORGANIZATION Georgia Institute of Tech. 6b. OFFICE SYMBOL (If applicable)	7a. NAME OF MONITORING ORGANIZATION U. S. Army Research Office
6c. ADDRESS (City, State, and ZIP Code) Atlanta, GA 30332	7b. ADDRESS (City, State, and ZIP Code) P. O. Box 12211 Research Triangle Park, NC 27709-2211
8a. NAME OF FUNDING/SPONSORING ORGANIZATION U. S. Army Research Office 8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER DAAG29-85-K-0023
8c. ADDRESS (City, State, and ZIP Code) P. O. Box 12211 Research Triangle Park, NC 27709-2211	10. SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO. / PROJECT NO. / TASK NO. / WORK UNIT ACCESSION NO.

11. TITLE (Include Security Classification)
The Role of Experience in Common-Sense and Expert Problem Solving

12. PERSONAL AUTHOR(S)
Janet L. Kolodner

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
Final	FROM 12/1/84 TO 11/30/87	March 1988	16

16. SUPPLEMENTARY NOTATION The view, opinions and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy, or decision, unless so designated by other documentation.

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	Problem Solving, Language Processing, Knowledge Acquisition, Reasoning Systems, Predictions

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

The objective of this research was to elucidate the role of experience in common-sense and expert problem solving. The aim was to discover and describe the processes involved in extracting useful conceptual knowledge from experience, in organizing and building the schemata to hold that knowledge, and in using that information in problem solving. In research areas as diverse as natural language processing and expert systems, researchers are plagued by the fact that the knowledge the systems need is hard to collect and input to the system. One way this bottleneck, called knowledge acquisition, can be relieved is by providing systems with a means of learning from their experiences. This research helps to lay the theoretical foundation for reasoning systems that (1) can become more expert through experience, (2) can make predictions and give advice based on previous experience in similar situations, and (3) can adapt to changes in their environments.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT ☐ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION Unclassified	
22a. NAME OF RESPONSIBLE INDIVIDUAL	22b. TELEPHONE (Include Area Code)	22c. OFFICE SYMBOL

DD FORM 1473, 84 MAR — 83 APR edition may be used until exhausted. All other editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

ARO 21697.11-MA

The Role of Experience in Common-Sense and Expert Problem Solving

Final Report

Janet L. Kolodner, PI
School of Information and Computer Science
Georgia Institute of Technology
Atlanta, GA 30332

ARO Proposal Number: 21697-MA
ARO Contract Number: DAAG29-85-K-0023
Period of Contract: December 1, 1984 - Nov. 30, 1987

1 Problem Statement

The objective of this research was to elucidate the role of experience in common-sense and expert problem solving. Our aim was to discover and describe the processes involved in extracting useful conceptual knowledge from experience, in organizing and building the schemata to hold that knowledge, and in using that information in problem solving. In research areas as diverse as natural language processing and expert systems, researchers are plagued by the fact that the knowledge the systems need is hard to collect and input to the system. One way this bottleneck, called knowledge acquisition, can be relieved is by providing systems with a means of learning from their experiences. This research helps to lay the theoretical foundation for reasoning systems that (1) can become more expert through experience, (2) can make predictions and give advice based on previous experience in similar situations, and (3) can adapt to changes in their environments.

2 Background

In the work done under this contract, we have focussed on a problem solving technique called *case-based reasoning* (Hammond, 1986, Kolodner & Riesbeck, 1986, Kolodner & Simpson, 1984, 1985, 1988, Kolodner, et al., 1985, Kolodner, 1983, 1985, 1987a, 1987b, Rissland, 1982, Simpson, 1985). In case-based reasoning, a problem solver remembers previous similar situations and uses what it remembers about those situations to solve its new problem. Noticing similarities between experiences allows a problem solver to solve problems more efficiently, while remembering similar situations that resulted in failure allows a problem solver to anticipate and avoid failures in solving a new problem.

Our investigation has been primarily in the task domain of mediation, a complex real-world domain. We have consulted with experts to find out how they solve problems in this domain, and we have constructed a serires of progressively more sophisticated computer programs that model some of the processes involved in mediation. Our programs, called the MEDIATOR and

88 4 040

the PERSUADER, take as input the demands made by both sides in various disputes. Based on knowledge of previous contractual agreements, the programs each classify the current dispute with respect to other disputes with which they are familiar, and suggest solutions. Simulated feedback from both parties forces each program to repair its initial suggestion to be more in line with the previously unknown demands of the disputing parties. Each program then remembers its experience so that in later cases it can take shortcuts in problem solving and avoid previously-made mistakes.

The MEDIATOR (Kolodner & Simpson, 1984, 1985, 1988, Kolodner et al., 1985, Simpson, 1985) solves resource disputes concerning one disputed object in a common-sense way. The PERSUADER (Sycara, 1985a, 1985b, 1987a, 1987b, 1987c) solves labor mediation disputes similarly to the way a human mediator does. While the MEDIATOR showed the usefulness of case-based reasoning for a complex task and showed several functions a case-based reasoner can perform, the PERSUADER shows how case-based reasoning can be used for partial satisfaction of several competing and conflicting goals, and shows an instance of case-based reasoning being combined with analytic methods.

An example from the MEDIATOR's domain will illustrate case-based reasoning. We assume our hypothetical reasoner starts with the "book knowledge" we expect a novice to have. Through experience, that book knowledge becomes more refined, its domain of applicability is learned, and the previously unrelated facts become related and therefore more useful. The example is in the domain of mediation of common-sense disputes. A failed mediation attempt triggers a need to explain the failure. A later episode, in a different domain, but with the same goal structure, causes reminding of the first episode, and through case-based reasoning, a prediction and advice about a proposed solution are given.

> Two sisters are quarrelling over an orange. Their mother surveys the situation and proposes that each sister take half of the orange. One of the sisters complains, since she wants to use the whole peel for baking. Realizing the real nature of the conflict, the mother suggests that the sisters divide the orange agreeably: one will take the fruit and eat it, while the other will take the peel and use it for baking.

Analysis of this example shows that while the mother thought that both sisters had the same goal, she was mistaken. Though their sub-goals were in conflict, their goals were not. Stepping back and considering the real goals rather than the manifest ones resulted in a goal concordance. The following shows how this analysis is transferred in understanding and making a prediction about another situation. Here, we imagine the mother reading the following story in the paper:

> Egypt and Israel both want possession of the Sinai. The US suggests they cut it down the middle. Both Egypt and Israel complain.

Analogy to the orange dispute allows her to conclude that possession of the Sinai is merely a subgoal, that the real goals of the two countries should be considered, and that a mutually-agreeable split based on those goals be sought. This interpretation of the Sinai Dispute is done by *case-based reasoning.*

3 Results

Case-based problem solving uses previous experiences to suggest means of solving new problems. Recall of a previous experience can aid in understanding the intracasies and focus of a new problem, generating a plan for resolution of the problem, and in case of failure, in explaining and remedying the failure and re-evaluating the case. Recall and application of knowledge gained in dealing with previous novel cases can cut down the amount of reasoning necessary to resolve a new problem and can prevent failures from being repeated. In essense, case-based reasoning involves recall of a previous case, focus on those parts of the previous case that can be helpful in solving the new problem, and analogical transfer and then modification of some portion of the previous case to solve the new problem.

Case-based inference, in the simplest case, requires the following steps (Kolodner, 1987b, Kolodner & Simpson, 1988):

1. Recall a previous case.
2. Focus on appropriate parts of the case.
3. Adapt the focused-on parts of the previous case to fit the new case.

Recall of a case is done by probing the case memory. This is usually done several times during problem solving. Our programs probe memory each time they have a new goal to achieve. As the problem to be solved gets better defined, more specific cases become available. Thus, several cases may be used in the course of solving a single problem. In general, memory returns several cases rather than just one. Thus, the recall step also involves a filtering step in which the best-matching case of those retrieved from memory is selected.

Because any case that is recalled can be quite large, a case-based reasoner must be able to focus on the parts of the previous case that will be helpful in solving the new problem. This can be done by using the goals of the problem solver with respect to the new case. In short, focus is directed at those parts of the previous case that achieved the goal analogous to the one that must be achieved for the new case.

Because no two cases match exactly, the solution to a previous case is not usually exactly applicable to the new case. Thus, a case-based reasoner is responsible for adapting the parts of the previous case to fit the new case. In the simplest problems, there is no adaptation, and this step is merely a transfer step. In some situations, the method by which the old solution was derived is transferred to the new case, in some situations, domain-specific adaptation heuristics are applied, and in some situations, domain-independent adaptation heuristics are used. The case-based reasoner in effect acts as a *hypothesis generator* during the focus step, proposing possible ways to achieve the problem solver's goals, and acts as *hypothesis adapter* in the third step, turning the coarse proposals made by a previous case into solutions applicable to the new problem.

3.1 The Case-Based Reasoning Paradigm: The MEDIATOR

The major contribution made by the MEDIATOR project was in defining the problem solving paradigm underlying case-based reasoning. The MEDIATOR (Kolodner & Simpson, 1984, 1985, 1988, Kolodner, et al., 1985, Simpson, 1985) resolves common sense disputes based on experience solving previous similar problems. By common-sense disputes, we refer to the kinds people run into from day to day. Children quarrelling over possession of objects, colleagues needing the same resource at the same time, and disputes encountered in reading the newspapers are just a few of kinds of disputes the program deals with. The MEDIATOR program, developed by Bob Simpson, begins with a semantic memory detailing the kinds of disputes it might encounter (e.g., physical, economic, and political) and a set of common mediation plans (e.g., one cuts the other chooses, split the difference, divide by parts). As it resolves disputes, it builds up an episodic memory organized by the concepts in its semantic memory. During processing, it first attempts recourse to previous experience to resolve a problem, and if no applicable experience is available, it uses default means (based on exhaustive search of alternatives) to resolve the problem. It learns based on feedback about the decisions it has made. If feedback is positive, it reinforces its belief that a particular type of plan is appropriate to a particular problem by storing the case and the plan used to resolve it. When it encounters later problems with features similar to one it has stored in memory, it will be reminded of that case and check to see if the plan used there was appropriate to its new problem. A positive experience may thus provide a shortcut in later problem solving. If feedback is negative, the MEDIATOR tracks down its error, fixes the knowledge that was responsible and attempts resolution of the problem a second time based on the new knowledge learned during feedback and the corrected knowledge that caused the previous error. When it finally resolves the problem satisfactorily, and stores the entire case in memory, later reminding of that case will (1) allow the problem solver to resolve a later similar problem without making the same mistakes a second time or (2) help the problem solver to figure out what went wrong when a similar failure occurs in the future.

There are several novel aspects to the MEDIATOR project. First, its model of problem solving includes not only the planning part of problem solving, but also problem understanding and follow-up based on feedback. Problem understanding must be included as part of problem solving because problems specifications are often incomplete and ambiguous. Follow-up procedures are necessary in order for learning to happen. If a problem was solved successfully, follow-up might only include indexing the case appropriately in memory so that it can be recalled in future similar circumstances. If some error occured as a result of problem solving, follow-up procedures include explaining the reason for the failure and recovering from it or figuring out how it could have been avoided. It is these follow-up procedures that allow a problem solver to learn from its experience.

Second, the MEDIATOR was the first implemented case-based reasoner and showed several uses of case-based reasoning during problem solving. As illustrated in the MEDIATOR, case-based reasoning can facilitate reasoning during any of the problem solving tasks listed above. During problem understanding, previous cases can aid in classifying a problem and elaborating it. During plan generation, case-based reasoning is used to choose planning policies, to devise skeletal plans, to choose the actions, objects, and characters that take part in the plan, and to generate predictions about the results of executing a plan. During follow-up, previous cases can aid in assigning blame for an error and in choosing a method of recovering from a mistake.

Third, the MEDIATOR showed how the appropriate parts of a previous case can be focussed on during case-based reasoning. Focus in the MEDIATOR is "demand driven", where demand is provided by the goal the problem solver is attempting to achieve or the task it is attempting to carrying out. When the problem solver is trying to classify a problem, it is the problem classification of the previous case that is focussed on. When it is attempting to derive a skeletal plan, it is the abstract plan from the previous case that it checks for applicability. Since transfer of information from one case to another derives from this focus, the analogical transfer of information from one case to another can also be said to be driven by the demands of the problem solver.

Fourth, the MEDIATOR has a well-articulated long term memory for experience. Problem solving experiences presented to the MEDIATOR are indexed in memory by those features which differentiate it from other experiences stored there. The memory organization is based on MOPs (Kolodner, 1984, Kolodner & Cullingford, 1986, Schank, 1982).

A fifth novel feature of the MEDIATOR is in its use of the same problem solving model to both solve domain problems (in this case, to resolve disputes) and to track down and fix failures in reasoning. It is able to do this because it treats both types of problems as first, classification problems, and then, plan instantiation problems. In solving domain problems, it thus seeks to classify disputes it encounters according to whether they are physical, economic, or political disputes during the understanding phase of problem solving. Each of these dispute types "knows" which types of plans are commonly useful to its resolution. Thus, classification allows pointers to potentially applicable canned plans, which must then be refined for the particular problem.

During failure resolution, the MEDIATOR treats the failure it has encountered as its new problem. During the understanding phase of failure resolution (explaining the failure), it attempts to classify the error (as, e.g., a classification error, an elaboration error, a particular kind of elaboration error, a plan refinement error). Each of those error classifications has remediation plans associated with it to fix the faulty knowledge or faulty reasoning rule. It thus fixes its errors by instantiating and refining a plan appropriate to the kind of error it encountered (e.g., one can fix elaboration errors by using an alternate elaboration rule or by asking the value of a feature from the user). In the same way previous experiences can provide shortcuts in problem solving, previous failures can provide shortcuts in error recovery (e.g., the orange dispute above). This method of failure recovery has potential in domains where the types of failures that may be encountered the of known ways of recovering from each can be specified.

3.2 Precedent-Based Reasoning: The PERSUADER

While the MEDIATOR showed the usefulness of case-based reasoning and pointed out some important aspects of problem solving, the PERSUADER provided a more in-depth investigation of the processes involved in transferring information from one case to another. The PERSUADER's transfer method is a specialization of case-based reasoning called *precedent-based reasoning.* Prededent-based reasoning is a method of deriving a solution to a new case by recalling one that is highly similar, computing the differences between the recalled and the new case, and based on those differences modifying or patching the old solution to fit the new situation. Case-based reasoning can also be used for this last step. Precedent-based reasoning, as implemented in the PERSUADER (Sycara, 1985c, 1987a, 1987d, 1988b), involves the following steps:

1. Recall a previous similar case to act as "precedent".

2. Do a "coarse adaptation" or "adjustment" of the results of the previous case to create a "ballpark solution" to the new problem. The ballpark solution takes only a set of coarse-grained features into account, but does not deal with details. It is meant to compensate for the dissimilarity between the recalled precedent and the "ideal" precedent, if it existed.

3. Evaluate the ballpark solution to see if it can achieve (or partially achieve) the set of goals it is designed to achieve given the context of the current problem. Three categories of knowledge are used here: more detailed knowledge about the problem itself, knowledge of the problem solving context (i.e., the environment in which the problem is being solved) and its effects on the situation, and knowledge of past failures in similar situations.

4. Using a set of task-&-domain-specific heuristics coupled with previous experience, do a detailed modification of the ballpark solution to create a solution that will work in the current problem solving context.

The PERSUADER uses case-based reasoning to resolve labor management disputes. Mediation, in these situations, is an iterative process. The mediator first attempts to ascertain the goals of the disputants, then attempts to construct a reasonable solution to the dispute. Often, the presentation of the "reasonable" solution to the disputants elicits additional constraints from the disputants about the problem, and the mediator is forced to modify the solution or construct a new solution to fit the better-defined problem. This process might go on for several cycles. When the mediator is sure that it/he/she has a full understanding of the problem and has created the best possible solution, a process of argumentation is used to persuade one or both disputing parties to agree to a proposed solution (Sycara, 1985a, 1985b, 1985d).

The PERSUADER uses precedent-based case-based reasoning for a variety of tasks: to create an initial solution to a dispute, to resolve impasses brought about by a disputant who will not agree to a proposed solution, and to derive arguments of persuasion that are used in an attempt to persuade a recalcitrant party to agree to a solution. For each of these, the particular features of a case that differentiate it from an "ideal" precedent are different (step 2), the particular features used for evaluation are different (step 3), and the set of task-&-domain-specific heuristics are different (step 4), but the general process remains the same.

The PERSUADER shows in detail how precedent-based reasoning works for a particular domain (labor mediation), and just as importantly, shows under what circumstances it breaks down and what can be done when that happens. When no cases are available, the program employs analytic methods, in this case an adaptation of utility theory formulations that we call "preference analysis" (Sycara, 1987a, 1987b) to mediate between goals and come up with a compromise solution. Any program that uses case-based reasoning will need some kind of "from-scratch" method when cases are not available, and one appropriate to the particular domain must be chosen. When a case is so atypical that neither precedent-based reasoning nor a from-scratch method of dealing with normal cases from a domain (in this case, preference analysis) can be used, some way of using domain-independent knowledge must be used. The PERSUADER uses "situation assessment" (Sycara, 1987a, 1987c, 1987d), a method of case-based reasoning in which domain-independent knowledge describing an analogous causal situation is used. Each is explained briefly below.

Preference analysis is a process that takes the relative utility of the goals of each of the disputants in a dispute into account to measure the potential for agreement to a proposed contract. It is used to come up with a contract if no precedent is available, to evaluate several potential contracts with respect to each other, and to judge which tradeoffs might be appropriate when everybody's goals cannot be fulfilled. It is reported in detail in (Sycara, 1987a, 1987b).

Even when cases are available, precedent-based reasoning methods may not be appropriate. This is the case when the new case is different from what is expected in ways that predict that the usual types of solutions won't work. In labor/management disputes this happens when the company is being mismanaged, when the union or the company have goals that are out of line with the norms, and several other times. The PERSUADER's way of dealing with this type of situation is to classify it by its goal/plan interactions (much as Schank suggests in his formulation of TOPs), and to use knowledge about dealing with those abstract kinds of situations to solve the problem (Sycara, 1987a, 1987c). For example, if the company is being mismanaged, one applies "mismanagement remedies" in coming up with a solution. One mismanagement remedy is to punish those who are doing the mismanagement by placing an overseer over them to make sure they will do things correctly in the future. In the labor/management domain, this might translate into placing union members on the board of directors. An interesting aside to this method is that while it is hard in general to specialize general strategies or remedies to specific new kinds of situations, once it has been done the case can be remembered and case-based reasoning can also help here.

The methodology used in the PERSUADER integrates analytic methods (preference analysis) with heuristic methods (precedent-based reasoning and situation assessment) to create a highly robust problem solver (Sycara, 1987a). There are several ways the heuristic and analytic methods interact. The analytic method provides a way to construct a solution when heuristic methods cannot be used. The heuristic methods support the analytic by providing necessary information that would be tedious to obtain otherwise. The analytic method provides a means to evaluate a solution constructed by heuristic methods. The integration of analytic and heuristic methods provide the following advantages for the PERSUADER:

1. The problem solver does not break down when heuristic methods fail.
2. The problem solver can flexibly apply the most natural solution method to each problem it encounters, sometimes using a variety of methods to solve a single problem as the problem evolves.
3. Heuristic methods can be used to construct a ballpark solution, while analytic methods can be used to refine it to a detailed level if heuristic methods are incapable of doing that.

The PERSUADER's model, as a whole, presents models of (a) resolution of multiple conflicting goals, (b) planning for partial goal satisfaction, (c) persuasive argumentation, and (d) integration of heuristic and analytic methods. As a model of conflict resolution, the PERSUADER suggests what the ingredients of resolution strategies must be. As a model for partial goal satisfaction, it has implications for human decision making. As a system that embodies a theory of persuasive argumentation, it presents a novel framework for the study of attitude and belief modification. It also demonstrates the usefulness of case-based reasoning in a variety of tasks necessary for problem solvin gin complex domains. The novelty of the research is not only that it addresses problems little studied before, but also that it addresses them in an integrated framework.

3.3 Other Issues in Case-Based Reasoning

There are several additional case-based reasoning issues that we have addressed over both of these projects: what gets transferred during case-based reasoning and what types of case-based reasoning processes do that transfer, anticipating and avoiding previously-made mistakes, and representing cases.

3.4 Transfer and Adaptation Processes in Case-Based Reasoning

There are several processes that we have identified for making case-based inferences (Kolodner, 1987b):

1. Transfer the solution that achieved the current goal in the previous case.
2. Transfer the solution that achieved the goal and modify it based on differences between the current and previous cases.
3. Transfer the inference method by which the previous goal was achieved.
4. Create an abstraction of the problem descriptions of the old and new caes, extend it to fit the solution to the previous case, apply the abstraction to the new case to create the framework for a solution, and refine that framework to fit the new case.

The process to be used depends on a number of considerations. Process 1 is used when the goal to be achieved can be achieved by choosing a single value or fully-instantiated frame. This method is simplest, and is employed by the MEDIATOR. Process 2 is precedent-based reasoning, employed by the PERSUADER. It is appropriate when there are several goals to be achieved simultaneously, when the previous solution integrates the achievement of several goals simulateously, or when the problem solver's goal is not one that is easily decomposable into non-overlapping parts.

Process 3 (Kolodner, 1986, 1987b) is useful when the details of the old and new cases are so different that no particular features of the old case can be transferred to the new, but the environmental factors (e.g., constraints) that would be used to choose a plan to achieve the current goal are similar. In this case, the inference method used previously is used to achieve the goal in the new case. While neither the MEDIATOR nor the PERSUADER use this method, it is implemented in another program, called JULIA, that plans meals. JULIA uses this method if, for example, it is asked to plan a vegetarian meal. Upon remembering a previous meal where the main course was chosen by selecting a main course central to the specified cuisine and then finding a vegetarian recipe for it, JULIA is able to choose a main course for another vegetarian meal with a different cuisine by this method.

Process 4 (Kolodner, 1987b, Shinn, 1988a, 1988b) is an analogy method. In this method, a mapping is made between the problems of the current and previous cases. This mapping is used to create a solution schema that describes both cases. This solution schema will be an abstraction of the two cases. It is then applied to the new case, creating an abstract solution which must then be refined. This method subsumes the other methods, as it can transfer a solution directly, transfer a solution method, or transfer an abstraction of a solution that is then modified. It is, however, a more time-consuming process, and one that we would want our automated reasoners to do only if the easier methods are not directly applicable.

3.5 Anticipating and Avoiding Mistakes

Previous failures serve several purposes during problem solving (Kolodner, 1987a, 1987b). They provide warnings of the potential for failure in the current case, and they may also provide suggestions of what to do instead. Analyzing the potential for failure in a new case, a necessary part of capitalizing on an old failure, may require the problem solver to gather additional information, thus causing the problem solver to change its focus of attention. A previous failed case that was finally solved correctly can help the problem solver to change its point of view in interpreting a situation if that is what is necessary to avoid potential failure.

Errors in reasoning can happen during any problem solving step. The problem might have been misunderstood initially, resulting in incorrect classification of the problem or incorrect inferences during the problem elaboration phase. Since problem understanding is an early part of the problem solving cycle, such misunderstandings and incorrect inferences propagate through to the planning phase, resulting in a poor plan. A problem might be understood correctly and all the necessary details known about it, but might still be solved incorrectly because poor decisions were made while planning a solution. In general, such errors are due to faulty problem solving knowledge. The problem solver might not have complete knowledge, for example, about under what circumstances a particular planning policy or plan step is appropriate. Finally, a problem might be solved correctly but carried out incorrectly by the agent carrying out the plan, or unexpected circumstances might cause execution to fail. Reminding of a case where any of these things happened warns the the problem solver of the potential for the same type of error in the new case. If the previous case was finally resolved correctly, details of its correct resolution are used to provide suggestions for solving the new problem correctly.

Any time the problem solver encounters a case with a previous problem, it considers whether there is the potential for that problem in the new case. This may cause it to refocus itself until the potential for failure is determined, and if such potential is determined and the problem solver has to retract decisions made previous to the current one, then it must remake any decisions dependent on those decisions. Such processing, of course, requires that the problem solver be integrated with a reason-maintenance system that keeps track of the dependencies among its decisions.

In short, the steps that must be followed to capitalize on a previous failure are (Kolodner, 1987a): (1) determine what was responsible for the failure, if possible (thismay already be recorded, and if not, some short amount of time is spent attempting toderive it), (2) direct reasoning focus tothe decision in the new problem that is analogous to the one that cause the failure in the previous one (this may be the one currently being focussed on or one that its correct solution is dependent on), (3) check for the potential for the same failure in the new case, either by seeing if the explanation of the previous failure holds in the new case or by checking the reasons why the previous decision was made and seeing if the same justifications might apply in the new case (this step may require additional information gathering), (4) if not, potential for error isn ot there, so return to the interrupted reasoning step and keep going, (5) if so, rule out the previous errorful decision as a possibility for the current case, and if the previous case was finally resolved correctly, determine if the decision made when it was resolved correctly is applicable tothe new case, (6) if so, use it as a suggestion for a case-based inference, (7) if step 2 redirected reasoning focus, then redo whatever decisions must be redone as a result (i.e., by following dependencies) and return to the reasoning step that was interrupted.

3.6 Representing Cases

There are several representational issues that we have had to address to define our case-based reasoning processes appropriately. First we discuss the representational structure of cases. Then we discuss the knowledge that needs to reside with the solution part of a case.

Cases have five parts to them (Kolodner, 1985, 1987b, Kolodner & Simpson, 1988): (1) the problem being solved, stated in terms of goals to be achieved, constraints on those goals, and other environmental factors that go into choosing a solution, (2) the solution to the problem, including the reasoning that was done to come up with the solution and a set of predictions of what to expect if the solution is carried out correctly, (3) feedback from the world about what happened as a result of carrying out the solution proposed in (2), (4) evaluation of that feedback, and (5) next problem solving steps taken as a result of that evaluation (e.g., another case).

Because much of the processing in case-based reasoning requires knowing why previous decisions were made, what other decisions previous decisions were dependent on, and what was responsible for previous failures, there must be both a representational system and a bookkeeping system that keep track of that knowledge. In the systems we are building, we store this knowledge with the solution part of each case. In short, each value recorded in the solution has a *value frame* associated with it (Kolodner, 1986, 1987a). Each time the problem solver makes a decision, it records its decision in the value slot of the value frame and also records what led to the decision. This might include an inference rule that was applied and the set of values it was applied to. Value frames include facets for the chosen value, other values that were suggested as alternatives but not chosen, ruled out values, conditions that were taken into account in choosing a value, and the inference rule or set of steps used to make the decision. The knowledge kept in value frames supports both transfer of reasoning method from one case to another and avoidance of previously-made mistakes.

4 Conclusions

Our studies of case-based reasoning are showing that exploitation of previous experience provides considerable advantage to a problem solver. While there is much support structure needed for a case-based reasoner to do its work (a memory for cases, a reason maintenance system to keep track of dependencies, value frames to keep track of justifications and past reasoning), case-based reasoning allows a problem solver to exploit its experience to take shortcuts in reasoning and to anticipate and avoid previously-made errors. This might ultimately allow us to build expert and common-sense problem solving systems that can learn from both their successes and their mistakes. Of course, there are many problems we have not addressed here that must still be addressed: how to best index cases so that the best ones are made available by the memory, which cases to keep in memory and how to organize them so the problem solver is not inundated with cases, how to choose the best of many cases provided by a memory, how to integrate the memory with the problem solver, how to integrate a case-based reasoner with other reasoners it needs to communicate with, processes for tracking down and explaining failures, processes for generalizing from both successful and failed problem solving experiences. While work is being done in each of these areas in research projects at Georgia Tech (see, e.g., Kolodner, 1983, 1985, Hinrichs, 1988, Turner, 1986, 1988, Shinn, 1988a, 1988b) and elsewhere (e.g., Alterman, 1986, Carbonell, 1983, 1986, Hammond, 1986, Kass, 1986, Rissland, 1982, Ross, 1982, Schank, 1982, Sycara, 1988a, 1988b), there is still considerable work to be done on all of these problems.

In 1984, when this project began, case-based reasoning was virtually unknown in the field of Artificial Intelligence. Partially as a result of the work done under this contract, case-based reasoning is becoming widely known within AI, and there is a great deal of interest in the use of case-based reasoning methods. In addition, Georgia Tech is now known as a leader in the area of case-based reasoning. While in 1984, our research group was composed of 4 students, we now have over a dozen students doing work related to case-based reasoning at Georgia Tech. This work is currently supported by the Army Research Institute for the Behavioral Sciences, the National Science Foundation, and Lockheed AI Center. Support from DARPA will begin in the next months.

5 References

1. Alterman, R. (1986). An adaptive planner. *Proceedings of AAAI-86.*

2. Carbonell, J. G. (1983). Learning by analogy: formulating and generalizing plans from experience. In Michalski, R. S., Carbonell,J.G. and Mitchell, T.M. (Eds.) *Machine Learning,* Tioga Publishing Co.

3. Carbonell, J. G. (1986). Derivational analogy in problem solving and knowledge acquisition. In Michalski, R. S., Carbonell, J. G., and Mitchell, T. M. (Eds.) *Machine Learning, Vol II,* Morgan-Kaufmann Publishers.

4. Hammond, K. J. (1986) *Case-Based Planning: An Integrated Theory of Planning, Learning, and Memory.* Ph.D. Thesis. Dept. of Computer Science, Yale University, New Haven, CT.

5. Kass, A. (1986). Modifying explanations to understand stories. *Proceedings of the Conference of the Cognitive Science Society.*

6. Kolodner, J. L. (1983). Towards an Understanding of the Role of Experience in the Evolution from Novice to Expert. *International Journal of Man-Machine Systems,* November, 1983.

7. Kolodner, J. L. (1984). *Retrieval and Organizational Strategies in Conceptual Memory.* Lawrence Erlbaum Associates, Hillsdale, NJ.

8. Kolodner, J. L. (1985). *Experiential Processes in Natural Problem Solving.* Report No. GIT-ICS-85/23. School of Information and Computer Science. Georgia Institute of Technology. Atlanta, GA.

9. Kolodner, J. L. (1986). Some little-known complexities of case-based inference. *Proceedings of the Third Annual Workshop on Theoretical Issues in Conceptual Information Processing*, April, 1986.

10. Kolodner, J. L. (1987a). Capitalizing on failure through case-based inference. *Proceedings of the Conference of the Cognitive Science Society*, 1987.

11. Kolodner, J. L. (1987b). Extending problem solver performance through case-based inference. *Proceedings of the Fourth International Machine Learning Workshop*, June, 1987.

12. Kolodner, J. L. and Cullingford, R. E. (1986). Towards a memory architecture that supports reminding. *Proceedings of the Conference of the Cognitive Science Society.*

13. Kolodner, J. L. and Riesbeck, C. (1986). *Experience, Memory, and Reasoning.* Lawrence Erlbaum Associates, Inc. Publishers, Hillsdale, NJ.

14. Kolodner, J. L., and Simpson, R. L. (1984). Experience and problem solving: a framework. *Proceedings of the Sixth Annual Conference of the Cognitive Science Society*, Boulder, CO., 2-9.

15. Kolodner, J. L. and Simpson, R. L. (1985). Using experience as a guide for problem solving. it Machine Learning Research. Kluwer Academic Press.

16. Kolodner, J. L. and Simpson, R. L. (1988). *The MEDIATOR: A Case Study of a Case-Based Problem Solver.* Report No. GIT-ICS-88/11 School of Information and Computer Science. Georgia Institute of Technology. Atlanta, GA, 1988.

17. Kolodner, J. L., Simpson, R. L., and Sycara-Cyranski, K. (1985). A Process model of case-based reasoning in Problem Solving. *Proceedings of IJCAI-85.*

18. Rissland, E. (1982). Examples in the legal domain: Hypotheticals in contract law. *Proceedings of the Fourth Annual Conference of the Cognitive Science Society.*

19. Ross, B. (1982). *Remindings and Their Effects on Learning a Cognitive Skill/* Cognitive and Instructional Sciences Series CIS-19. Palo Alto, CA: Xerox Palo Alto Research Centers.

20. Schank, R. C. (1982). *Dynamic Memory: A Theory of Learning in People and Computer.* Cambridge University Press, London.

21. Shinn, H. (1988a). Abstractional Analogy. Submitted to AAAI-88.

22. Shinn, H. (1988b). Analogical mapping for case-based reasoning. Submitted to the 1988 Conference of the Cognitive Science Society.

23. Simpson, R.L. (1985). *A Computer Model of Case-Based Reasoning in Problem Solving: An Investigation in the Domain of Dispute Mediation.* Ph.D. Thesis. Technical Report No. GIT-ICS-85/18. School of ICS, GA Institute of Technology, Atlanta, GA.

24. Sycara, E. (1985a). Persuasive argumentation in resolution of collective bargaining impasses. *Proceedings of the Seventh Annual Conference of the Cognitive Science Society*, August, 1985.

25. Sycara, E. (1985b). Arguments of persuasion in labor mediation. *Proceedings of IJCAI-85*, August, 1985.

26. Sycara, E. (1985c). *Precedent-Based Reasoning in Expert Labor Mediation.* Report No. GIT-ICS-85/22. School of Information and Computer Science. Georgia Institute of Technology. Atlanta, GA.

27. Sycara, E. (1985d). *Arguments of Persuasion: Two Papers.* Report No. GIT-ICS-85/19. School of Information and Computer Science. Georgia Institute of Technology. Atlanta, GA.

28. Sycara, E. (1987a). *Resolving Adversarial Conflicts: An Approach Integrating Case-Based and Analytic Methods.* Ph.D. Thesis. Report No. GIT-ICS-87/26. School of Information and Computer Science. Georgia Institute of Technology. Atlanta, GA.

29. Sycara, E. (1987b). Utility theory in conflict resolution. *Annals of Operations Research: Approaches to Intelligent Decision Support*, 1987.

30. Sycara, E. (1987c). Finding creative solutions in adversarial impasses. *Proceedings of the Conference of the Cognitive Science Society.*

31. Sycara, E. (1987d). Planning for Negotiation: A Case-Based Approach. *Proceedings of the DARPA Knowledge-Based Planning Workshop*, Austin, TX, December 1987.

32. Sycara, E. (1988a). Resolving Goal Conflicts via Negotiation. Submitted to AAAI88.

33. Sycara,E. (1988b). Using Case-Based Reasoning for Plan Adaptation and Repair. *Proceedings of the DARPA Workshop on Case-Based Reasoning.* May, 1988.

6 Publications and Technical Reports

1. Kolodner, J. L. (1985). *Experiential Processes in Natural Problem Solving.* Report No. GIT-ICS-85/23. School of Information and Computer Science. Georgia Institute of Technology. Atlanta, GA.

2. Sycara, E. (1985). Persuasive argumentation in resolution of collective bargaining impasses. *Proceedings of the Seventh Annual Conference of the Cognitive Science Society*, August, 1985.

3. Sycara, E. (1985). Arguments of persuasion in labor mediation. *Proceedings of IJCAI-85*, August, 1985.

4. Sycara, E. (1985). *Precedent-Based Reasoning in Expert Labor Mediation.* Report No. GIT-ICS-85/22. School of Information and Computer Science. Georgia Institute of Technology. Atlanta, GA.

5. Simpson, R. L. (1985). *A Computer Model of Case-Based Reasoning in Problem Solving: An Investigation in the Domain of Dispute Mediation.* Ph.D. Thesis. Report No. GIT-ICS-85/18. School of Information and Computer Science. Georgia Institute of Technology. Atlanta, GA.

6. Sycara, E. (1985). *Arguments of Persuasion: Two Papers.* Report No. GIT-ICS-85/19. School of Information and Computer Science. Georgia Institute of Technology. Atlanta, GA.

7. Kolodner, J. L. and Simpson, R. L. (1985). Using experience as a guide for problem solving. *Machine Learning Research.* Kluwer Academic Press.

8. Kolodner, J. L. and Riesbeck, C. (1986). *Experience, Memory, and Reasoning.* Lawrence Erlbaum Associates, Inc. Publishers, Hillsdale, NJ.

9. Kolodner, J. L. and Simpson, R. L. (1986). Case-Based problem solving: A framework. Experience, Memory, and Reasoning.

10. Kolodner, J. L. (1986). Some little-known complexities of case-based inference. *Proceedings of the Third Annual Workshop on Theoretical Issues in Conceptual Information Processing*, April, 1986.

11. Sycara, E. (1987). *Resolving Adversarial Conflicts: An Approach Integrating Case-Based and Analytic Methods.* Ph.D. Thesis. Report No. GIT-ICS-87/26. School of Information and Computer Science. Georgia Institute of Technology. Atlanta, GA.

12. Sycara, E. (1987). Utility theory in conflict resolution. *Annals of Operations Research: Approaches to Intelligent Decision Support*, 1987.

13. Sycara, E. (1987). Finding creative solutions in adversarial impasses. *Proceedings of the Conference of the Cognitive Science Society.*

14. Kolodner, J. L. (1987). Capitalizing on failure through case-based inference. *Proceedings of the Conference of the Cognitive Science Society*, 1987.

15. Kolodner, J. L. Extending problem solver performance through case-based inference. *Proceedings of the Fourth International Machine Learning Workshop*, June, 1987.

16. Sycara, E. (1987). Planning for Negotiation: A Case-Based Approach. *Proceedings of the DARPA Knowledge-Based Planning Workshop*, Austin, TX, December 1987.

17. Kolodner, J. L. and Simpson, R. L. (1988). *The MEDIATOR: A Case Study of a Case-Based Problem Solver*. Report No. GIT-ICS-88/11 School of Information and Computer Science. Georgia Institute of Technology. Atlanta, GA, 1988.

18. Sycara, E. (1988). Computer Aiding in Negotiation (Extended Abstract). Submitted to 2nd Conference on Computer Supported Cooperative Work.

19. Sycara, E. (1988). Multi-Agent Compromise via Negotiation (Extended Abstract). Submitted to the 8th Workshop on Distributed AI.

20. Sycara, E. (1988). Resolving Goal Conflicts via Negotiation. Submitted to AAAI88.

21. Sycara,E. (1988). Using Case-Based Reasoning for Plan Adaptation and Repair. *Proceedings of the DARPA Workshop on Case-Based Reasoning*. May, 1988.

7 Scientific Personnel and Advanced Degrees Awarded

1. Janet L. Kolodner, PI

2. Robert L. Simpson, Ph.D. student – not directly supported by the contract; however, computer time for his programming efforts (he programmed the MEDIATOR) and time spent by the PI advising his research were charged to this contract. Simpson received a Ph.D. in June, 1985 based on his work developing and describing the MEDIATOR and the problem solving paradigm it illustrated. He is now a Program Director at DARPA. Ph.D. Thesis: *A Computer Model of Case-Based Reasoning in Problem Solving: An Investigation in the Domain of Dispute Mediation.*

3. Ekaterini (Katia) Sycara, Ph.D. student. Received her Ph.D. in June, 1987 based on her work on the PERSUADER. Her graduate work was almost completely funded by this project. She is now a Research Associate in the Carnegie-Mellon University Robotics Institute. Ph.D. Thesis: *Resolving Adversarial Conflicts: An Approach Integrating Case-Based and Analytic Methods.*

4. T. Rangarajan, Ph.D. student – supported for two quarters in 1985. Was later dismissed from the project for lack of progress.

5. Thomas Hinrichs, MS student, Ph.D. student – supported for three quarters in 1986 and 1987. Worked on representing cases. Received his MS in 1987, partially funded by this project. He is now a Ph.D. student working on the use of case-based and other problem solving techniques to solve problems in open-worlds, Ph.D. expected in August, 1989.