Incremental Reminding: the Case-based Elaboration and Interpretation of Complex Problem Situations

Brian M. Slator and Ray Bareiss

The Institute for the Learning Sciences
Northwestern University, Evanston, IL 60201
{slator,bareiss}@ils.nwu.edu

Abstract

When solving a complex problem, gathering relevant information to understand the situation and imposing appropriate interpretations on that information are critical to problem solving success. These two tasks are especially difficult in weak-theory domains -- domains in which knowledge is incomplete, uncertain, and contradictory. In such domains, experts may rely on experience for all aspects of problem solving. We have developed a case-based approach to problem elaboration and interpretation in such domains.

An experience-based problem-solver should be able to incrementally acquire information and, in the course of that acquisition, be reminded of multiple cases in order to present multiple viewpoints to problems that present multiple faults. We are addressing issues of 1) elaboration and interpretation of complex problem situations; 2) multiple interpretations; and 3) the role of categories as the foci of reasoning in the context of the Organizational Change Advisor (ORCA). Its model of incremental reminding is a plausible mechanism for this sort of expert problem solving behavior, and one that works well in weak theory domains. Because there is an implicit cost associated with retrieving a complex case, ORCA implements a retrieval time similarity function that requires both general expectations and specific situational relevance be considered before a story is told to the user; this increases the chances that a retrieved case will be useful.

1. Introduction

A "weak-theory" domain, such as business or law, is characterized by a lack of reliable general principles: knowledge is incomplete, uncertain, and even contradictory (Porter, Bareiss, and Holte, 1990). Expert problem solving in such contexts often involves much more than routinely gathering data and generating answers; it involves interpreting a complex problem situation in, perhaps, many ways. Experts answer questions and tell stories; they explore alternative hypotheses and implement intermediate solutions; they also gather further data and revise their assessments, and then incrementally produce new and improved solutions to partially solved problems.

Case-based reasoning (CBR), has been proposed as an effective means of elaborating and interpreting a complex situation (e.g.. Simpson, 1985). In this context, elaboration means acquiring the features necessary to form an interpretation, and interpretation amounts to categorizing a situation as an instance of a known problem. A case serves as a specific model[1] for interpreting a situation and tells the problem-solver which features to attend to (out of a potentially huge range of possibilities) and how important their presence (or absence) might be.

Most implementations of CBR have implicitly assumed that the use of a single retrieved case is sufficient to solve a problem. If another case needs to be retrieved, it is because the current one proved to be inappropriate. This is similar to the single fault assumption in diagnosis (see, for example, MEDIATOR: Simpson, 1985; Kolodner et al., 1985; CHEF: Hammond, 1989).[2] However, there are many complex problem situations that would seem to demand multiple interpretations where, because of the presence of multiple faults, a single retrieval will not do. To account for these situations, we make a weaker assumption: the non-interacting problem assumption,

This research was supported in part by the Defense Advanced Research Projects Agency, monitored by the Air Force Office of Scientific Research under contract F49620-88-C-0058 and the Office of Naval Research under contract N00014-90-J-4117, by the Office of Naval Research under contract N00014-89-J-1987, and by the Air Force Office of Scientific Research under contract AFOSR-89-0493. The Institute for the Learning Sciences was established in 1989 with the support of Andersen Consulting, part of The Arthur Andersen Worldwide Organization. The Institute receives additional support from Ameritech, an Institute Partner, and from IBM.

[1] We use the term model in the sense of Weiss (Casnet; 1978) or Nii et al. (Sonar interpretation; 1982) rather than in the qualitative reasoning sense that the term often now connotes in AI.

[2] Ashley and Rissland (HYPO; 1987) do not attempt to select the single most closely matching case but rather to retrieve all cases which match (or nearly match) the current case on any relevant underlying dimensions; Redmond (CELIA; 1990) does not in principle, but does in published examples; Hinrichs and Kolodner (1991) retrieves a number of cases, then these are decomposed and pieces of several might be employed in a synthesized solution.

i.e., that multiple faults do not interact in such a way that the diagnostically significant features of any problem are masked.

This paper describes a system which implements a multiple-retrieval CBR approach to problem interpretation and elaboration based on iteratively developing a picture of the problem situation. Rather than entering a script-based initial data-gathering dialog to create a featural description of the problem situation, we retrieve a case that will form a solid beginning, and then successively elaborate the problem description through comparison with a sequence of cases retrieved in response to the results of previous comparisons. In other words, a picture of the problem situation is constructed incrementally by comparing and contrasting it with multiple stored cases; and as more stored cases are retrieved, more is learned about the problem, and the interpretations become more "on point".

A typical way to tackle multiple retrieval case-based reasoning is through "difference link" refinement. In these systems (e.g. Protos: Bareiss 1989; Julia: Hinrichs and Kolodner, 1991, and MEDIATOR: Simpson, 1985; Kolodner et al., 1985), this requires a memory in which cases are relatively indexed by significant differences, in order to cache the results of multiple retrievals during problem solving, to provide a shortcut in a similar future episode. With difference links, acquired cases are finely distinguished from each other and problem solving is accomplished by categorization, which is achieved by traversing difference links until an acceptably matching case is located.

However, there are special problems posed when the "weak theory" domain is extremely broad and complex, and the number of distinct cases is large. In this event, the sheer combinatorial magnitude of creating difference links is prohibitive, since evaluation of all the pairwise possibilities simply cannot be accomplished. For domains of this type another strategy must be implemented. Rather than traverse difference links, which are unlikely to exist if they have not been extensively pre-enumerated (because it is unlikely that a similar enough situation was previously encountered), we make a series of retrievals based on the featural differences of the problem situation and the retrieved cases.

2. The Problem/Task Domain

We are studying these issues in the context of organizational change consulting. An "organizational change" consultant is typically contacted by a company when a significant event has occurred that is outside of the company's expertise. The consultant is hired to assess the state of the company, to diagnose its problems, and then to recommend and, typically, to implement changes in the company's structure or way of doing business. This is an extremely complex problem, and in a weak-theory domain such as organizational change, there is no substitute for experience; even though past experiences may not exactly mirror a client's situation, they provide useful analogies illustrating solutions tried in previous situations and how well they have worked.

A knowledge-based system to assist change management consultants should: 1) encourage systematic exploration of a complex problem by asking relevant, context-sensitive questions; 2) suggest necessary information to acquire from the client; 3) propose hypotheses about the client's problems when evidence of relevance is uncovered; 4) present analogies to relevant past cases from the consulting firm's corporate memory; 5) offer assistance in solving the client's problems by providing actions and outcomes associated with past cases.

3 ORCA: Organizational Change Advisor

The Organizational Change Advisor (ORCA; Bareiss and Slator, 1992) is a consulting aid and advice giving program that interacts with a consultant to gather information about a client, and then tells stories that are found to be analogous to the client's situation. The goal of the ORCA system is to build a picture of a complex situation and then to interpret it by categorizing it in several different ways. To do this, ORCA gathers information about a client's situation by posing questions to the consultant. The answers to these questions build up a description in memory which, in turn, reminds ORCA of previous, similar cases, whose features provide expectations that translate to additional questions to ask. This elaboration leads to several different retrievals of previous cases, and these are used to build a description of the client's situation through a series of "follow-up discussions" that incrementally contribute to the further elaboration and building of the client case.

ORCA implements an algorithm for case-based interpretation to retrieve seemingly appropriate cases on the basis of weak remindings. To do this, ORCA operates over a memory of stories and domain elements connected to each other with "reminding" links (where reminding is a heuristic association between domain element such that finding one increases the likelihood of finding the other: i.e. the traditional notion of predictive indexing). ORCA employs the strategy of looking for the important features of the case in order to elaborate the problem situation. If they are found, the new situation can be interpreted as an instance of a known type of experience.

As the user answers questions about the relevance of domain elements to the client's situation, ORCA uses the answers to manage a queue of possibly relevant stories and, when sufficiently reminded of a particular story, shows it to the user as a case related in some way to the client's case. As new information is gathered, either from user input or as a consequence of the user revising their assessment of the client's case, the reminding network considers other types of problems and proposes further stories to the user. In this way,

reminding produces expectations which are used to form queries, and it is the success or failure of these expectations that enables interpretation and efficient categorization.

3.1 Representation

ORCA's cases, gathered both from interviewing expert consultants and from searching professional journals, are intended to help the user consider realistic problem solving alternatives and to familiarize the user with the cases in memory most closely related to their client's situation. Unlike many case-based reasoners, however, cases are stories to be presented for use by a human user rather than fully represented entities for use in autonomous problem solving. The sole representation of a case is indexical; that is, the only case features accessible to the system are those used in making retrieval decisions. The bulk of the case, including its problem solving advice is stored as a block of text (or, in some cases, video) that is opaque to the system.

ORCA's memory contains both descriptive features of business situations and inferable abstract problem descriptions. The features are drawn from a vocabulary of descriptors of business situations developed by the consulting firm. These include direct observables, such as "a change in senior personnel has taken place" and reasonably straight-forward inferences, such as "friction exists between organizational units." The abstract problem descriptions include a set of concepts taken from the consulting firm's methodology and another set of common sense descriptions borrowed from the study of conventional, proverbial, wisdom. These abstract descriptions provide an explicit way of organizing the problem-describing features, and a way of interpreting the cases from the firm's corporate memory which are indexed by those features. ORCA represents categories extensionally as sets of retained cases. Rather than being exemplars of a single abstract problem type, ORCA's cases contain features relating to several different abstract problem types. In this way ORCA encodes the corporate reality that, to make a medical analogy, every patient typically has many diseases.

The abstract problem descriptions are based on an indexing strategy that encodes a notion of common sense societal wisdom (Owens 1990). For example, everyone knows that problems arise when an employee is made to report to more than a single boss: conflicting orders are given, time is wasted negotiating priorities, and in the worst cases, the bosses are dissatisfied and the employee is frustrated. Situations like this are so well-known that, over the centuries, society has developed and preserved a shorthand system of aphorisms to describe them. In a case like this, an impartial witness might observe, "No man can serve two masters" or, more picturesquely, "A pig with two masters will starve." In ORCA, proverbial expressions of this sort provide a framework of categories for organizing domain features and interpreting cases.

The basic index in ORCA's memory is the reminding link. ORCA also makes use of censor links to suppress remindings, but has no difference links (see Bareiss, 1989 for further description of these link types). The confirmatory link is a new index type that has been introduced to link abstract problem categories with features which tend to be confirmatory and hence, in general, worth pursuing when reminded of the category. During problem solving, determining the presence of a confirmatory feature in the client's situation suggests the relevance of presenting category instances to the user. In other words, confirmatory features are reasonable, general, things to ask about when reminded of a category.[3]

3.2 Elaboration and Interpretation

The role of a case in elaborating and interpreting a new situation is that the case serves as a specific model for acquiring features and imposing an interpretation. Under this scheme, an initial, possibly weak, reminding leads to an hypothesis which causes a retrieval. The retrieved case is used as a model for interpretation, providing expectations of additional features the case should possess. When the expectations are met, this provides an interpretation (i.e. a classification of the problem situation), and in any event, because cases are multiply classified, new remindings are produced which yield new hypotheses as possible interpretations to be confirmed.

ORCA has a two-level hierarchy of models: MOPs representing abstract problem types and cases representing particular experience. The typical flow of reasoning is that the system is reminded of an abstract problem type, asks questions to confirm its relevance, then retrieves an appropriate exemplar. No particular confirmatory feature is necessary to confirm the relevance of a category; however, one or more must be present. This method is in place because of the relatively high cost of working through a case; therefore, we require confirmation in terms of the norms of the category and in terms of superficial similarity, to assure the case is worth discussing.

Cases have prototypicality ratings with respect to abstract problem types. These qualitative ratings --- strong, medium, and weak --- partially order the cases as exemplars of the corresponding abstract problem types. When presented with a client situation, ORCA tries 1) to systematically acquire a picture of the situation by asking relevant questions, 2) to classify the situation repeatedly, as newly acquired information suggests relevant abstract problem types, and 3) to exemplify those problem types to the user by presenting similar past cases. The primary difficulty is that every situation that ORCA encounters will embody a multiplicity of

[3]These features are the "norms" of the category (Schank, 1982; Kolodner, 1984; Riesbeck and Schank, 1989).

problems. As a consequence, problem solving involves reasoning from a multiplicity of partially matching cases. ORCA is faced with solving a classification problem in which the cases are quite complex and the diagnostic assessment tools are weak, subjective, inconsistent, and inconclusive.

Classification is an iterative process of forming and revising a set of active hypotheses as the user answers ORCA's questions. At the beginning of an ORCA session, the user is asked to identify one or more significant business problems in the client situation, chosen from a pre-enumerated list of problems (i.e. merger or restructuring.) These are connected via reminding links to memory elements representing features and abstract problem descriptions. Each element of which ORCA is reminded is placed on a "best-first" agenda, and questions to confirm each, ordered by importance, are added to a question agenda. At each step in the problem solving process, ORCA asks the question at the head of the agenda.

Answering a question places its associated feature into a confirmation set, activates the associated remindings, and may cause all of ORCA's agendas to be re-evaluated. Generally, such a re-evaluation occurs as a result of a reminding link that associates the newly acquired feature with one or more abstract problem types. For example, if the user confirms that "the company's R&D staff is small," ORCA will be reminded of an abstract problem characterized as "he who looks not ahead looks behind." This reminding will cause ORCA to seek confirming features for the abstract problem, and the questions "Does the client neglect R&D in its planning and budgets?" and "Does the organization use outmoded equipment?" will be placed at the front of the question agenda.

As the features and proverbs are confirmed, the cases indexed by these elements are placed on their own agenda. Cases that come to the front of the agenda may be presented to the user when a heuristic estimation of their relevance exceeds a dynamically computed threshold of reminding strength. Prototypicality ratings suggest the relevance of cases to abstract problem types, but because of the extreme diversity of instances of abstract problem types, they are not the sole determinant of which case is presented. When the problem situation is interpreted to a point where ORCA is reminded of a category, cases for consideration are selected on the joint basis of prototypicality and featural match to the client's situation. No matter how prototypical a case is, however, it will not be presented unless there is some degree of featural match; for example, a story about a bank, and another about a taco stand, might both fall into the category exemplified by the proverb "he who looks not ahead looks behind." -- but a financial institution with the same problem will almost certainly profit less from hearing about the taco stand than from hearing the story of the bank. This two-level system imposes requirements to suppress distant analogies that may not be of great utility to a consultant.

After a case is chosen and its story is told, a follow-up dialog is entered in which the user is asked to elaborate the client's situation by comparing it to the various features of the story. (Some of these elements are already known to be true of the client, of course, since the story was shown on this basis to begin with.) The follow-up dialog amounts to listing the important descriptive elements associated with the story and asking which of these are reasonably associated with the client case. This comparison affords the user the opportunity to think about the most relevant features of their client in the specific context of a previous case. The presentation of past cases aids the user in determining the likelihood of various problems, provides advice by analogy, and provides tangible contexts for acquiring additional information about the client by prompting the user to make explicit comparisons. As discussed earlier, features acquired through these elaborative dialogs remind ORCA of additional possible interpretations of the situation.

4. An Example of ORCA in Action

The objective of an ORCA consultation is to create a case representing the client's situation. This is an iterative picture that develops over days and weeks, as the consultant is able to gather information about the client. At first, the consultant is only expected to answer general or surface-level questions, and often the proper answer to a question is "I don't know." Over time, by following up on ORCA's suggestions, the consultant will find the answers to specific questions to provide a detailed picture of the client's organization and its problems. Imagine that a staff consultant is called into his manager's office, and she tells him only the following basic information:

> *A manufacturing plant renowned for its innovation has encountered difficulty in maintaining its culture since competitive pressure forced the introduction of a "just-in-time" manufacturing system. The plant originally was organized around semi-autonomous production teams, who had been previously allowed to manage their own work at their own individual rates.*
>
> *Though some problems stem from the new system, most arise from the style of the new plant manager. Under pressure from corporate headquarters to use a system designed by an outside consulting firm, he severely limited employee involvement in the process. Employees received neither involvement in the design of the just-in-time system, nor any explanation of the cost problem facing the plant.*

The consultant begins his task by consulting ORCA; he enters the client's name and chooses one or more "change drivers" from a menu of eight (e.g., high-level

problem types such as merger/acquisition, business relocation, restructuring/reorganization, competitive threats, strategic planning, political crisis, and so on). In this instance, the consultant enters the client's name and two change drivers: reorganization/restructuring and competitive threats.

Change drivers are initial entry points into ORCA's memory that are directly connected to some of its memory elements. Choosing a change driver reminds ORCA of both surface features and abstract problem types, and causing confirming questions to be placed, best first, on the question agenda. ORCA then poses each of these to the user in turn. The combination of the two change drivers, "reorganization/restructuring" and "competitive threats", reminds ORCA of a number of abstract problem types. The strongest remindings are to problem types exemplified by proverb-23: "Who looks not before finds himself behind.", proverb-17: "He that cannot adapt is obsolete", and proverb-21: "You've got to spend money to make money". ORCA puts questions on the question agenda to attempt to confirm these problems.

The first question ORCA asks is "Has the industry experienced major technological improvements in recent years?" The user answers "Yes" to this question (which confirms feature-1138). This question was suggested by feature-27: "The organization exists in a highly competitive field" which was itself confirmed by the change driver "Competitive Threats". One of ORCA's initial hypotheses, articulated by proverb-17, "He that cannot adapt is obsolete", is confirmed by this answer because of a confirmatory link between feature-1138 and proverb-17. As a consequence of the interaction thus far, ORCA chooses a story to tell. The story is judged sufficiently relevant because it has surface similarity to the problem situation (through feature-27 and feature-1138), as well as abstract similarity (through proverb-17). ORCA tells the following story:

> *Kodak Cuts Management - Lower Managers Take Control (abridged)*
>
> *At Eastman Kodak Co.'s apparatus division, which makes parts for Kodak printers, copiers, and film processors, the management team was cut by 30% and layers slashed from seven to three. But some first-line supervisors, who suddenly had to set goals and strategy for their products rather than just carry out orders, "couldn't make it" despite a training course, says Frank Zaffino, the division's vice president and general manager. "They were used to being star technicians, not communicators or leaders." [...]*

After a story has been told, ORCA engages the user in a follow-up dialog in order to elaborate the client situation by assessing the degree to which the particular features of the story are relevant to the client. In this instance the story is relevant in some respects, but the cause of the problem is off point: rather than losing autonomy, as in the client's case, the workers at Kodak have difficulty handling increased autonomy. In the follow-up dialog, the features are presented in order of increasing relevance to the story, in the expectation that the most productive contrast can be found in this way.

Through a follow-up dialog, it becomes apparent the two situations differ in significant ways. In particular, ORCA asks, "In order to achieve its vision, does leadership consider downsizing, severe restructuring, or a reallocation of resources imperative?" This feature is weakly relevant to the Kodak story, but the user rates it highly relevant to his client. Then ORCA asks "Is a new vision, image or style planned?" This feature is also weakly relevant to the Kodak story, but the user rates it as moderately relevant to the client. Then ORCA asks, "Is the organization unwilling to risk implementing new systems, processes, or projects? This feature is rated as moderately relevant to the Kodak story but, crucially, the user sees their client in exactly the opposite way and rates this feature as having a negative correlation to the client's problem: the client is more than willing to risk implementing new systems, in fact this seems to be a big part of the problem. Based on the newly acquired information gathered as a consequence of the follow-up interaction, ORCA indicates that another story is immediately available. This story is only moderately prototypical of proverb-17: "He that cannot adapt is obsolete", but it is strongly linked to proverb-23: "Who looks not before finds himself behind.", which is the hypothesis that ORCA has chosen to explore next .

> *New Zealand Government Struggles with Change (abridged)*
>
> *The Inland Revenue Department in New Zealand had an interesting situation in that they had some very elderly technology which was not really servicing their needs at all well. They did a major information plan to decide on the technology of the future and ended up with a plan which would involve them over a four year time period basically reinstalling every single system in the place [...]*

This story goes on to relate the difficulties a government agency has trying to install a new computer system over the objections of the workforce. The story is more on point than the Kodak story, because it describes how this resistance was overcome through better informing the workforce about planned changes.

The ORCA consultation continues in this way, from story to story, as long as the user believes it to be worthwhile to answer questions and view cases. At the end of the session, the user is given a report that includes a summary of all sessions to date including a list of the features and abstract categories that are confirmed for the client, a list of questions to ask the client (i.e., questions that the user could not answer during the current session), synopses of all cases presented, and other relevant information. Armed with this report, the user is able to gather further information

and to use that information to answer more of ORCA's questions in future sessions.

ORCA is able to assimilate the client's case into memory because there is a sense in which the consultant is telling the client's story by answering the questions ORCA poses. At the end of the consulting engagement, because of the answers given, the new client case is indexed in exactly the same way as the pre-existing cases in ORCA's memory. The final task of the consultant, at the end of the consulting engagement, is to provide a narrative account of the engagement to associate with the case built for the client. Then the new case can be added to ORCA's permanent knowledge base and become a part of the corporate memory that ORCA will use to reason about future clients.

5. Conclusion

Case-based reasoning provides an effective means of reasoning about a complex problem situation because it relies on recalling actual past experiences as models for elaborating and interpreting complex problem situations. Incremental elaboration and interpretation is necessary for case-based reasoning in complex, weak-theory domains because there is no "correct answer" assumption -- the problem situations are complex and information becomes available only in the course of problem solving.

The development of ORCA has been motivated by three theoretical issues that we are exploring computationally. The first issue concerns the nature of effective strategies for incremental elaboration and interpretation of complex problem situations. The second is multiple interpretations of a problem situation, perhaps based on multiple, partial views. The third is the role of categories as the foci of reasoning, i.e., the model that a problem solver is reminded of a category and confirms the reminding by retrieving a similar exemplar.

However, interpreting a complex problem situation, albeit through multiple retrievals, incremental refinements, elaboration, and the telling of relevant stories, is only a part of the problem solving process. The harder part, and the more interesting component of modeling expert behavior, is to follow through and find a way to automatically coalesce the advice from multiple cases into a coherent, unified plan. This remains as one of the major open problems waiting to be solved. In the meantime, ORCA provides a useful tool for cooperative human-computer problem solving.

6. Acknowledgments

ORCA is being built at the Institute for the Learning Sciences by Mike Engber, Kerim Fidel, Andy Gordon, Jim McNaught, Leena Nanda, Tamar Offer-Yehoshua, Josh Tsui, and John Welch in collaboration with Beth Beyer, Jeff Davies, Marita Decker, Dan Halabe, and Maureen Zabloudil of Andersen Consulting. Roger Schank and Chris Riesbeck have contributed valuable ideas throughout the process; Tom Hinrichs helped out by commenting on an earlier draft.

7. References

Ashley, K.D. and Rissland E.L. (1987). Compare and Contrast, a test of expertise. *Proceedings of AAAI-87*

Bareiss, R. (1989). *Exemplar-based Knowledge Acquisition: a Unified Approach to Concept Representation, Classification, and Learning*, San Diego: Academic Press

Bareiss, R, and Slator, B.M. (1992). From Protos to ORCA: reflections on a unified approach to knowledge representation, categorization, and learning. (ILS Technical Report #20).

Hammond, K.J. (1989). *Case-based Planning: Viewing Planning as a Memory Task*. San Diego: Academic Press

Hinrichs, T.R. and Kolodner, J.L. (1991). The roles of adaptation in case-based design. *Proceedings of AAAI-91*.

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

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*

Nii, H.P., Feigenbaum, E.A., Anton, J.J., Rockmore, A.J. (1982). Signal-to-Symbol Transformation: {HASP/SIAP} Case Study, *The AI Magazine*, pp. 23--35, Spring.

Owens, C. (1990) Indexing and Retrieving Abstract Planning Knowledge. PhD thesis, Yale University, Department of Computer Science.

Porter, B., Bareiss, R. and Holte, R. (1990) Concept Learning and Heuristic Classification in Weak-Theory Domains. *AI Journal*, 45, 229-263.

Redmond, M. (1990). Distributed Cases for Case-Based Reasoning; Facilitating Use of Multiple Cases. *Proceedings of AAAI-90*.

Riesbeck, C.K. and Schank, R.C. (1989). *Inside Case-Based Reasoning*, NJ: Lawrence Erlbaum Assoc.

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

Simpson, R.L. (1985). *A Computer Model of Case-Based Reasoning in Problem Solving: An Investigation in the Domain of Dispute Mediation*. PhD Dissertation, Georgia Institute of Technology.

Weiss, S.M., Casimir, A.K., Amarel, S. (1978). A Model-Based Method for Computer-Aided Medical Decision-Making, *Artificial Intelligence*, 11, pp. 145--172