Integrating Case- and Rule-Based Reasoning

Soumitra Dutta
INSEAD, Fontainebleau, France
Piero P. Bonissone
General Electric Corporate R & D, Schenectady, New York

ABSTRACT

Rule-based reasoning (RBR) and case-based reasoning (CBR) have emerged as two important and complementary reasoning methodologies in artificial intelligence (AI). For problem solving in complex, real-world situations, it is useful to integrate RBR and CBR. This paper presents an approach to achieve a compact and seamless integration of RBR and CBR within the base architecture of rules. It is shown that the integration of CBR and RBR is possible without altering the inference engine of RBR. The paper focuses on the possibilistic (interpreted on the basis of similarity) nature of the approximate reasoning methodology common to both CBR and RBR. In CBR, the concept of similarity is cast as the complement of the distance between cases. In RBR the transitivity of similarity is the basis for the approximate deductions based on the generalized modus ponens. Approximate reasoning under uncertainty is also incorporated into the integration and is useful for dealing with many real-life situations and providing a comprehensive representation for CBR. This integration is illustrated in the financial domain of mergers and acquisitions. These ideas have been implemented in a prototype system, called a Mergers and Acquisitions Reasoning System (MARS).

KEYWORDS: ***case-based reasoning, rule-based reasoning, plausible reasoning, possibility theory, financial application of knowledge-based systems***

1. INTRODUCTION

In this section, we introduce rule- (R) and case-based reasoning (CBR) methodologies, describe some important related issues, emphasize the need for their integration, and outline the focus and structure of the paper.

Address correspondence to Soumitra Dutta, INSEAD, Boulevard de Constance, 77305 Fontainebleau Cedex, France.

Received May 1, 1992; accepted September 29, 1992.

International Journal of Approximate Reasoning 1993; 8:163–203

655 Avenue of the Americas, New York, NY 10010 0888-613X/93/$6.00

1.1 Rule- and Case-Based Reasoning

Rule-based reasoning [1] is one of the most popular reasoning paradigms used in artificial intelligence (AI). The reasoning architecture of rule-based systems has two major components: the *knowledge base* (usually consisting of a set of "IF... THEN..." rules representing domain knowledge) and the *inference engine* (usually containing some domain-independent inference mechanisms, such as *forward/backward chaining*). The general solution of RBR is shown in Figure 1. Given an input problem, applicable rules are first found by matching against the rules of the knowledge base; then, intermediate results are generated by the chosen inference mechanism (such as forward or backward chaining), and the process is repeated till the desired solution state is reached. The chosen inference mechanism (forward/ backward) determines whether the *antecedents* (forward) or the *consequents* (backward) of the rules in the knowledge base are used for matching and whether the desired solution state is the attainment of a particular conclusion (forward) or the determination of the existence of certain data (backward). The knowledge base contains the domain knowledge pertinent to the problem, and the solution is, in general, found by incrementally exploring the rule graph formed by the rules in the knowledge base.

Case-based reasoning (or analogical reasoning), though common and extremely important in human cognition, has only recently emerged as a major reasoning methodology. This type of reasoning involves solving new problems by identifying and adapting *similar* problems stored in a library of past experiences/problems. The reasoning architecture of CBR consists of a case *library* (stored representations of previous experiences/problems

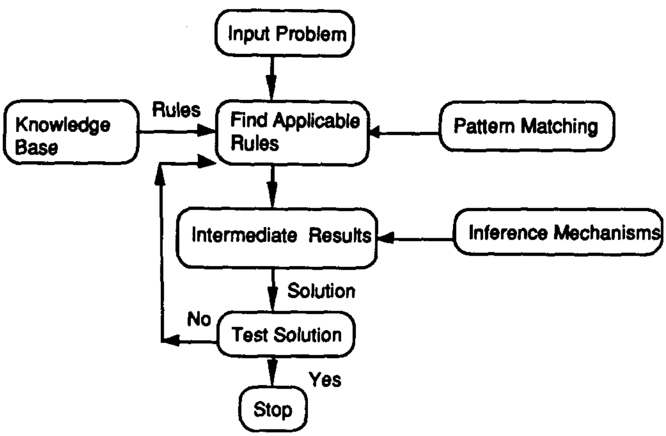

Figure 1. Solution structure in RBR.

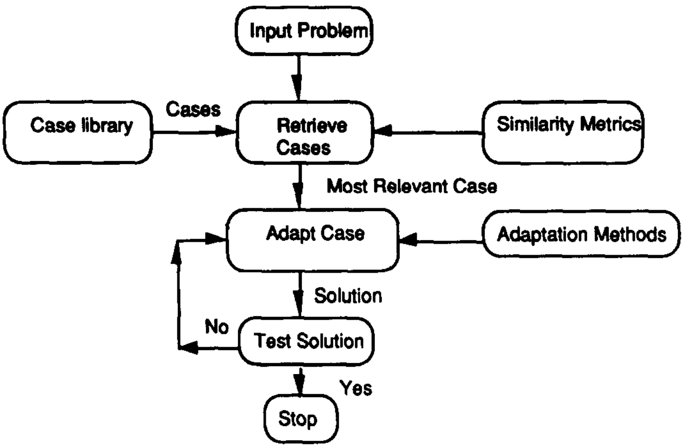

Figure 2. Solution structure in CBR.

solved) and an *inference cycle*. The important steps in the inference cycle of CBR are to *find* and *retrieve* cases from the case library that are most relevant to the problem at hand (input) and *adapt* the retrieved cases to the current input. This has been illustrated in Figure 2. Within this broad framework, two major classes of CBR can be identified [2]: *problem-solving* CBR and *precedent-based* CBR. In problem-solving CBR, the emphasis is on adapting the retrieved cases to find a plan or a course of action to solve the input problem (such as in industrial design and planning [3]). In contrast, the focus in precedent-based CBR is to use the retrieved cases to justify/explain an action/solution (common in legal reasoning [4]).

Comparison of Figures 1 and 2 provides a useful perspective on RBR and CBR. The general solution structures of both RBR and CBR are quite similar: in both some matching (against either rules or cases) is performed and then some procedures (inference mechanisms or adaptation techniques) are used iteratively to generate the desired solution. This hints to the fact that it may be possible to seamlessly integrate RBR and CBR; however, there are some important differences between RBR and CBR, as typically discussed in the literature. The contents of the case library are, in general, more complex structures than simple IF...THEN...rules, and a variety of representational schemes (such as frames and memory organization packets (MOPS) [5]) have been used in the literature to represent their complexity. Pattern matching in CBR is usually more involved than simply matching the left-hand side (LHS) (*antecedents*) or the right-hand side (RHS) (consequents) of rules (in RBR). This complexity is caused by the data structures used to represent the case library and the fact that the match of the input problem with the various cases in the case library is usually partial. Unlike RBR, where a solution is obtained by incrementally searching the rule graph, CBR typically generates a complete solution (the selected case from the case library) first, and then progressively adapts it to solve the problem at hand. Although there are well-defined procedures (such as backward or forward chaining) for searching a rule graph, there are no equally well-defined procedures for case adaptation as it is usually ill structured and thus less susceptible to generalization.

1.2 Issues in RBR and CBR

From Figures 1 and 2 and the above description, one can identify several important issues in RBR and CBR. We outline below some of the issues that are most relevant to this paper and explain our treatment of them in later sections (4.2 and 4.3).

1.2.1 RULE-BASED REASONING Issues of concern in RBR are

- **Nature of Facts and Rules** In conventional RBR, facts and (IF ...THEN...) rules are strictly *categorical* in nature (e.g., IF *it is cloudy*, THEN *it shall rain*). Both commonsense knowledge and domain expertise are more naturally expressed in terms of plausible rules (e.g., IF *it is cloudy*, THEN *it is POSSIBLE that it may rain*). Conventional IF...THEN... rules also require a strict Boolean match on the *premises* and the *conclusions*; however, this is very restrictive as real-world situations are often fuzzy and do not match exactly with rule premises and conclusions. For example, the premise it is *cloudy* is fuzzy and many a time the sky can be classified as *partially cloudy*. Thus, it is necessary to be able to accommodate partial degrees of matching in rule premises and conclusions.
- **Inference in Rule Graph** Given the above stated need to incorporate uncertainty in the structure of rules, some mechanisms have to be used to propagate the partial degrees of matching through the rule graph to determine the degrees of confirmation and refutation of the various conclusions. These mechanisms include means to aggregate the uncertainty of the premises, propagate this uncertainty to the conclusions and aggregate the uncertainty of the conclusion. The last task is important because a particular conclusion may be reached via multiple proof paths, each path contributing to the confirmation or refutation of the conclusion. Different uncertainty calculi are described in the literature to deal with some or all of these aspects.
- **Structure of Knowledge Base** Large knowledge bases can contain many (hundreds or thousands of) rules, and under such conditions, it is necessary to structure the knowledge base appropriately for the

purposes of efficiency in inference and ease of debugging and knowledge engineering. *Partitions*, *abstraction hierarchies*, and *contexts* are some of the techniques used in the literature for structuring the knowledge base. *Belief revision* mechanisms are also required for maintaining the validity and consistency of the knowledge base.

1.2.2 CASE-BASED REASONING Important issues in CBR are

- **Representation** This refers to the representational scheme used for the cases in the case library. There are several dimensions along which this issue can be analyzed. The data or memory structure for cases popular in the literature are frames or some variant of a frame system, such as MOPS [31]. Cases can also be stored in an *interpreted* or *uninterpreted* format. In an interpreted format, all relevant features and event sequences are explicitly stored; whereas in an uninterpreted format, some features and events have to be derived later. The description of cases in the case library can also be *complete* or *partial*.
- **Structure of Case Library** In practice, the number of cases in a case library can be vary large. It is important to be able to efficiently index and search through the case library to retrieve the most relevant case. A variety of techniques are used in the literature, including *flat structures* (with serial or parallel searches) and many variants (*shared feature/prioritized/redundant* [31]) of discrimination nets. The selection of surface features and indices play an important role here.
- **Need** Case-based reasoning is useful only for solving certain kinds of problems (e.g., legal reasoning) and under certain conditions (e.g., when domain rules are expensive and hard to formulate). It is important to be able to identify this need for CBR and then either activate or deactivate CBR in a problem solver.
- **Retrieval and Similarity Metrics** While determining which case to retrieve from the case library, it is important to get not just the most *similar* case, but the most *relevant* case. Simple *static* evaluation metrics (such as predetermined weights) can often lead to surprising results. More sophisticated *dynamic* metrics which take into account such factors as prior experiences, anomalies [6], expectations, and the particular goal/task under consideration generally yield better results. The description of uncertainty and imprecision (given above for rule premises in RBR) is also true here. Matches between the input problem and the cases are usually partial.
- **Adaptation** This is perhaps the least structured and most difficult part of the CBR solution. There are few general guidelines. Adaptation methods can be classified into two broad categories: *structural*, that is, directly modify the solution from the retrieved case and *derivational*, that is, rerun the solution procedure of the retrieved case.

Specific procedures used in the literature include *parameter manipulation*, *critic-based adaptation* and *reinstantiation* [5].

- **Integration** Much of the current research in CBR considers it in isolation from other reasoning techniques; however, in general, CBR is just another proof (solution) path to a goal (problem). It is important to consider the interactions of CBR with other reasoning methodologies that might also be employed in a problem solver.

Although the issues of *representation*, *structure*, *retrieval*, *similarity metrics*, and *adaptation* have been discussed in the literature, the issues of *need* and *integration* have received comparatively negligible attention. This is because most prior research in CBR has considered CBR in isolation and has ignored its interaction with other reasoning methodologies. It is only recently that some researchers (see Section 1.4) have begun exploring the interactions of CBR with other reasoning methodologies. Figure 3 conceptually illustrates how CBR would be integrated into a reasoning system which might employ other reasoning methodologies (such as RBR and/or model-based reasoning) besides CBR. Note that there is now a need for *aggregating* the solutions obtained via CBR and other reasoning techniques.

1.3 Uncertainty in RBR and CBR

It is evident from the above description that uncertainty and imprecision are pervasive in the reasoning cycle of deductive (rule-based) and analogical (case-based) reasoning systems.

In rule-based reasoning, uncertainty and imprecision are present in both the domain knowledge (*plausible rules* and *fuzzy/imprecise* premises) and the inference techniques used to search the rule graph.

In CBR, uncertainty and imprecision are present in the semantics of *abstract features* used to index the cases, in the evaluation and (hierarchical) aggregation of the *similarity* measures computed across these features,

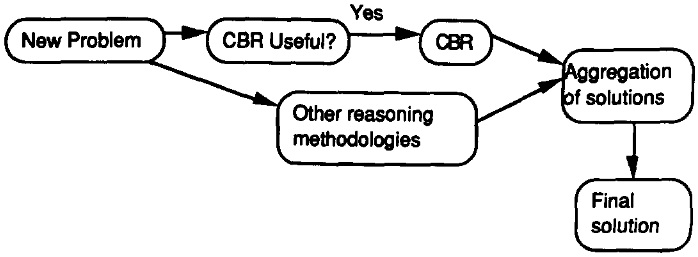

Figure 3. Integrating CBR with other reasoning methodologies.

in the determination of *relevancy* and *saliency* of the similar cases, and in the solution adaptation phase.

In Section 2.2 we will show how most of this uncertainty can be modeled by using fuzzy predicates and plausible rules to derive abstract features from the surface features. Similarity measures can be defined as the complement of metrics between fuzzy-sets (cases). The similarity measure can be aggregated or chained (using the transitivity of similarity) according to well-defined operators (Triangular norms [*T-norms*]).

1.4 Integration of Reasoning Methodologies

The need to integrate diverse reasoning techniques for effectively solving complex real-world problems has been recently recognized by the AI community. The integration of CBR with other reasoning methodologies is being researched by many researchers, including Carbonell and Velose [7, 8] (integration of CBR and classical search problem solvers), Hammond and Hurwitz [9] (integrating CBR and explanation-based reasoning), Goel [10] (integrating CBR and model-based reasoning), Braverman [11] (integrating CBR and explanation-based learning), Rissland and Skalak [2], Dutta and Bonissone [12], Branting [13], and Golding and Rosenbloom [14] (integrating RBR and CBR).

RBR and CBR are largely *complementary* reasoning methodologies. RBR can better represent specialized domain knowledge in a modular, declarative fashion, whereas CBR can better represent past experiences and domain complexity [5]. Significant benefits are possible by combining RBR and CBR. For example, CBR can directly enhance RBR by providing a context for screening the knowledge base and extending the coverage of rules by representing exceptions (to the rule) in the form of cases. Going the other direction, RBR can enhance CBR by expressing domain knowledge to dynamically determine the contextually dependent relevance of a feature set (or attributes of a case) to a given goal and dynamically select the best similarity/relevancy measure to use for case retrieval. There are numerous domains in which it is important to combine RBR and CBR, for example, the legal [15, 16] and financial domains [12] (see Section 3.3 for an example).

1.5 Focus and Structure of Paper

This paper is concerned with the integration of RBR and CBR in an uncertain and dynamic world. Rather than "patching together" two different types of representational and reasoning frameworks, we have chosen to attempt the integration within one architectural framework, namely that of RBR. RBR has been chosen as the base architecture as it is still the most popular and commercially successful AI reasoning methodology. As shown in later sections, it is possible to achieve a compact, seamless integration of the two reasoning methodologies without changing the inference engine of RBR. This has, as discussed later, some advantages over other architectures (see Section 5.1). The incorporation of uncertainty into the reasoning framework gives the system added power in handling real-world situations, which are almost invariably uncertain and dynamic. It also leads to a more accurate treatment of CBR, as it is inherently a nondeductive form of approximate reasoning in which there is significant uncertainty and imprecision, for example, in the semantics of the case features and in determining the similarity/relevancy of prior cases to the input problem/goal. The significance of our work arises from the fact that though RBR and CBR are two extremely important reasoning methodologies, there has been very little research in combining the two.

The domain chosen for the illustration of our ideas is the financial domain of Mergers and Acquisitions (M & A). M & A represent a real-world situation, which is complex, uncertain, dynamic, and relevant for business today. The ideas and technical approach detailed in this paper have been implemented in a prototype system, called *a Mergers and Acquisitions Reasoning System* (MARS) [17].

This paper contains four other sections. Section 2 illustrates the role of approximate reasoning techniques and contrasts probabilistic and possibilistic (similarity-based) reasoning systems. A brief introduction to the domain of M & A is provided in Section 3. The need for integrating RBR and CBR in this domain is also explained in that section. Section 4 provides an overview of MARS, illustrates the nature of similarity-based reasoning in MARS, and then describes the integration of RBR and CBR in MARS. Section 5 compares our work with related research and concludes the paper by describing the contributions and limitations of this research.

2. APPROXIMATE REASONING SYSTEMS

The task of a reasoning system is to determine the *truth value* of statements describing the state or the behavior of a real-world system; however, this hypothesis evaluation requires complete and certain information, which is typically not available. Therefore, approximate reasoning techniques are used to determine a *set of possible worlds* that are logically consistent with the available information. These possible worlds are characterized by a set of propositional variables and their associated values. As it is generally impractical to describe these possible worlds to an acceptable level of detail, approximate reasoning techniques seek to determine some properties of the set of possible solutions or some constraints on the values of such properties [18–21].[1]

A large number of approximate reasoning techniques have been developed over the past decade to provide these solutions (see references [22, 23] for a survey). These techniques can be roughly subdivided into two basic categories according to their *quantitative* or *qualitative* characterizations of uncertainty. Among the quantitative approaches, we find two types of reasoning that differ in the semantics of their numerical representation. One is the *probabilistic reasoning* approach, based on probability theory. The other one is the *possibilistic reasoning* approach, based on the semantics of many-valued logics. We will briefly contrast these two types of quantitative representations and focus our discussion on possibilistic reasoning systems. A good survey of fuzzy set-based approaches to approximate reasoning is given in reference [24].

2.1 Probabilistic and Possibilistic Reasoning Systems

Probability-based reasoning, or *probabilistic reasoning* seeks to describe the constraints on the variables that characterize the possible worlds with conditional probability distributions based on the evidence in hand. Their supporting formalisms are based on the concept of *set-measures*, additive real functions defined over certain subsets of some space.

These methods focus on chance of occurrence and relative likelihood. They are oriented primarily toward the choice of decisions that are optimal in the *long-run*, as they measure the *tendency* or *propensity* of truth of a proposition without assuring its actual validity. Thus, probabilistic reasoning estimates the frequency of the truth of a hypothesis as determined by prior observation (objectivist interpretation) or a degree of gamble based on the actual truth of the hypothesis (subjectivist interpretation).

Possibilistic reasoning, which is rooted in fuzzy set theory [25] and many-valued logics, seeks to describe the constraints on the possible worlds in terms of their *similarity* to other sets of possible worlds.

These methods focus on *single* situations and cases. Rather than measuring the tendency of the given proposition to be valid, they seek to find another proposition that resembles (according to some measure of similar-

[1] The authors want to acknowledge Enrique Ruspini's private communication, which is the basis for the content of this section. The interested reader should consult references [35–38] for further elaboration on this topic.

ity) the hypothesis of interest but that is valid. Thus, possibilistic reasoning asserts that a related, similar (and usually less specific) hypothesis is true.

It should be noted that there are other interpretations [24] for possibilistic reasoning which are not based on possible-world semantics. Of special interest is Zadeh's interpretation of a possibility distribution "as a fuzzy restriction which acts as an elastic constraint on the values that may be assigned to a variable [26]. This interpretation, in conjunction with the linguistic variable approach [27], is the basis for the development of numerous fuzzy control applications [28].

As a final comment, it should be noted that there are situations in which it is useful to consider the simultaneous representation of probability and possibility. Such cases have been analyzed by Zadeh in the definition of the probability measure of fuzzy events [29] and by Smets in the extension of belief functions to fuzzy sets [30]. Given the duality of purpose and characteristics between probabilistic and possibilistic methods, these technologies ought to be regarded as complementary rather than competitive [31].

2.2 Similarity-Based (Possibilistic) Reasoning

Given the purpose and characteristics of probabilistic and similarity-based possibilistic reasoning, it is clear that these technologies ought to be regarded as being complementary rather than competitive.

The single-case orientation of similarity-based possibilistic techniques makes them particularly suitable for case-based reasoning. In CBR, it is typically the case that the problem in hand (probe) has never been encountered before. The inference in CBR is based on the existence of cases *similar enough* (i.e., close enough) to the probe to justify the adaptability of their solution to the current problem.

The notion of similarity is based on the concept of *metric* or distance, as opposed to that of set measure. Distances are functions that assign a number greater than zero to pairs of elements of some set (for sake of simplicity, we will assume the range of this function to be the interval $[1, 1]$). Distances are *reflexive*, *commutative*, and *transitive*. *Similarity* can be defined as the complement of distance, that is,

$$S(A, B) = 1 - d(A, B).$$

The basic structural characteristics of the similarity functions is an extended notion of transitivity that allows the computation of bounds on the similarity between two objects A and B on the basis of knowledge of their similarities to a third object C:

$$S(A, B) \geq T(S(A, C), S(B, C))$$

where T is a Triangular-norm [32–35]. Any triangular norm $T(A, B)$ falls in the interval

$$T_\omega(A, B) \leq T(A, B) \leq \mathrm{Min}(A, B)$$

where

$$\begin{aligned} T_\omega(A, B) &= \mathrm{Min}(A, B) && \text{if } \mathrm{Max}(A, B) = 1 \\ &= 0 && \text{otherwise} \end{aligned}$$

$T_\omega(A, B)$ is referred to as the *drastic* T-norm (to reflect its extreme behavior) and is clearly noncontinuous. By changing one of the axiom of the T-norms [34, 35], we can derive a subset of T-norms, referred to as *copulas*, such that any copula $T(A, B)$ falls in the interval

$$\mathrm{Max}(0, A + B - 1) \leq T(A, B) \leq \mathrm{Min}(A, B).$$

Thus, we can observe that the if we use the lower bound of this range in the expression describing the transitivity of similarity, we obtain the triangular inequality for distances. If we use the upper bound, then we obtain the ultrametric inequality.

This similarity notion is a direct extension of the notion of equivalence that is at the root of the theory of rough sets, which can be captured by the modal logic S5. This notion is further described by Ruspini in references [18–21]. In summarizing Ruspini's results, we can observe that the notion of accessibility captures the idea that whatever is true in some world ω, is true, but in a modified sense, in another ω' that is accessible from it. When considering multiple levels of accessibility (indexed by a number between 0 and 1), this relation, measuring the resemblance between two worlds, may be used to express the extent by which considerations applicable in one world may be extended to another world.

The basic inferential mechanism, underlying the *generalized modus-ponens* [27], makes use of inferential chains and the properties of a similarity function to relate the state of affairs in the two worlds that are at the extremes of an inferential chain.

We have briefly summarized the semantics of similarity-based possibilistic reasoning, its role in determining the similarity between possible worlds (cases), and its mechanism to propagate similarities through a reasoning chain (rule chain). On this basis, we have established a common ground upon which we can build the integration of CBR and RBR. Before proceeding to describe such an integration, we need to justify the reasons for integrating these two methodologies. This motivation will be provided by the description of the problem domain of Mergers and Acquisition, which is used to test the integration.

3. MERGERS & ACQUISITIONS

This section introduces the domain of M & A and emphasizes the need for integrating RBR and CBR in M & A.

3.1 Introduction and Overview

The structure of corporate USA has been changed dramatically by the flood of mergers and acquisitions witnessed over the past years. Today, a flurry of mergers are sweeping through European industry as it prepares for 1993. Annually, these deals total tens of billions of US dollars. The average M & A deal is enormously complex and involves sophisticated reasoning and planning on the part of several parties. To lend some useful conceptual abstraction, we can consider two players of interest in simple M & A deals: the *raider* (who usually initiates a takeover attempt) and the *target* (which is the company of interest to the raider). Another player of interest who is outside the structure of the actual M & A deal, but has a keen interest in the entire process is the *professional arbitrageur* (who tries to make arbitrage profit by wisely shifting his investments during the merger process). Although the actions of each of these players vary from deal to deal, it is possible to identify certain basic actions associated with their individual roles. For example, some of the representative actions of a raider are

- **Target Monitoring** The raider has to track possible takeover targets by constantly evaluating different aspects of the various companies in accordance with his or her own goals, which could be *benevolent* (acquire, integrate, and manage the target company) or *malign* (bust the target for quick profits).
- **Target Evaluation and Selection** The raider has to select a particular company as the target. This is usually done with extreme care as the entire process of a M & A deal is very risky and expensive, both in time and money.
- **Merger Strategy Selection** The raider has to select the desired strategy of attack once a target has been selected. This involves complex reasoning and planning and is dependent on the raider's perception of the reaction of the management and stockholders of the target company.
- **Target Response Evaluation and Attack Strategy Modification** The target's response (*acceptance* or *defiance*) to the raider's merger offer determines how the raider subsequently modifies the chosen attack strategy.

Even in simple M & A deals, other complicating factors, such as multiple bidders and legal complications, often arise. The reader may consult the references [36–38] for more details on various aspects of M & A.

3.2 RBR and CBR in Mergers & Acquisitions

The above section provided a glimpse of the complex reasoning and planning required on the part of the various players in a M & A deal. Though a variety of reasoning and problem-solving methodologies are required for every step in the M & A deal (see reference [39] for details), RBR and CBR play a crucial role in the process.

RBR can be used to represent the domain expertise that is required for structuring various aspects of the M & A deal or deciding upon the next best course of action. For example, the raider may have certain expertise (rules) to determine which company to select as a possible target (this is similar to a classical *diagnostic* problem).

CBR also plays an extremely important role in the M & A process. For example, the prior experiences of other companies subject to hostile takeovers often guide the target's evaluation of and response to a hostile takeover bid by the raider. Even a cursory glance over a professional guide to mergers and acquisitions (e.g., see reference [40]) reveals the extensive use of prior cases for justifying strategies or explaining certain actions. Within the entire range of activities in a M & A deal, CBR is perhaps most useful for the legal and tax-related aspects. We explain below in some detail how both RBR and CBR are required for reasoning about one aspect of a M & A deal: the *anti-trust* defense.

3.3 The Anti-trust Defense

Usually, when a raider launches a hostile takeover attempt, the target has to devise an elaborate defense strategy. Michel and Shaked [37] note that "anti-trust arguments are one of the most frequently used forms of merger defense." The laws governing anti-trust cases depend on several merger guidelines (e.g., the 1982 and 1984 guidelines) issued by the US Department of Justice (DOJ). Much of the reasoning involved in the interpretation and application of guidelines regarding anti-trust laws can be expressed by rules. For example, the 1982 guidelines specified that markets where the postmerger HHI (a mathematical measure of market concentration) was above 1800 were *highly concentrated* and if the post-merger HHI was between 1000 and 1800, then the market was *moderately concentrated* and so on; however, such rules by themselves are not enough as [37] "it is not possible to remove the exercise of judgement from the evaluation of mergers under the anti-trust laws." This exercise of judgment is predominant in resolving such issues as definition and measurement of market, efficiency arguments and treatment of foreign competition. This is where CBR can help and is used extensively.

For example, consider the $5.1 billion attempt by Mobil to takeover Marathon on November 1, 1981. Marathon began the takeover defense by filing an anti-trust lawsuit against Mobile (if successful, then Mobil would become the largest marketer of gasoline in the USA with an estimated 10% market share). The key issue here was whether section 7 of the Clayton Act (which provides that "no person... shall acquire... stock... where, in any line of commerce... in any section of the country, the effect of such acquisition may be substantially to lessen competition") was being violated by the merger. Judge J. M. Manos of the Ohio Court ruled against Mobil and in his ruling [37] "referred to past cases similar to the Mobil–Marathon situation. In the 1962 case of Brown Shoe Company, the combined market share of Brown Shoe and G. R. Kinney Co. was found to exceed 5% nationwide, rising to 57% in some cities. In the 1966 case of Pabst Brewing Company, the merged firm would have a combined market share of 4.49% in the USA and up to 23.95% in Wisconsin." In all these three cases, section 7 of the Clayton Act was found to be violated.

This brief example illustrates the important role that cases and rules have in the M & A domain and stresses the need for their integration.

3.4 Computer Tools for M & A

Personal contacts and human communication perhaps play the most important role in determining the outcome of a M & A deal; however, a variety of computer tools are also available commercially for assisting decisions regarding various aspects of a M & A deal. These tools use quantitative (financial) data and conventional statistical models to generate historical and forecasted financial statements, balance sheets, financial ratios, accounting combinations, and other similar results. Typically, these tools do not provide support for other useful features such as intelligent information retrieval (from on-line news services), RBR (expert knowledge), CBR, uncertainty management, and dynamic planning (see [17]). Below we introduce MARS, our prototype system, which attempts to provide support for such facilities in an integrated environment.

4. MARS: A MERGERS & ACQUISITIONS REASONING SYSTEM

In this section, we first provide a quick overview of MARS and then focus on the integration of RBR and CBR within MARS.

4.1 Overview of MARS

MARS [17] is a prototype AI reasoning system that both simulates and provides expert advice regarding the actions of the raider, the target, and the arbitrageur. The general software architecture of MARS is as shown in Figure 4. There are four independent simulators. The global simulator provides a simulation of the variations of the macroeconomic variables affecting the M & A deal (e.g., the interest rate and the T-Bill price). The three other simulators simulate the reasoning and planning strategies of the raider, target, and the arbitrageur, respectively. There is a fusion of different reasoning techniques in all four simulators and each of them is independently capable of integrated reasoning and planning with uncertain, incomplete, and time varying information.

The first version of MARS was implemented in Common LISP using KEE and Reasoning with Uncertainty Module (RUM) [41, 42], and ran on the Symbolics workstation. A second version of MARS has been implemented recently using PRIMO [42] and is capable of being executed on the sun workstations also. The knowledge base of MARS is frame based and consists of approximately 550 KEE units. Figure 5 shows the user interface for MARS. The windows to the left in Figure 5 depict details about the various companies in MARS (there can be more than three

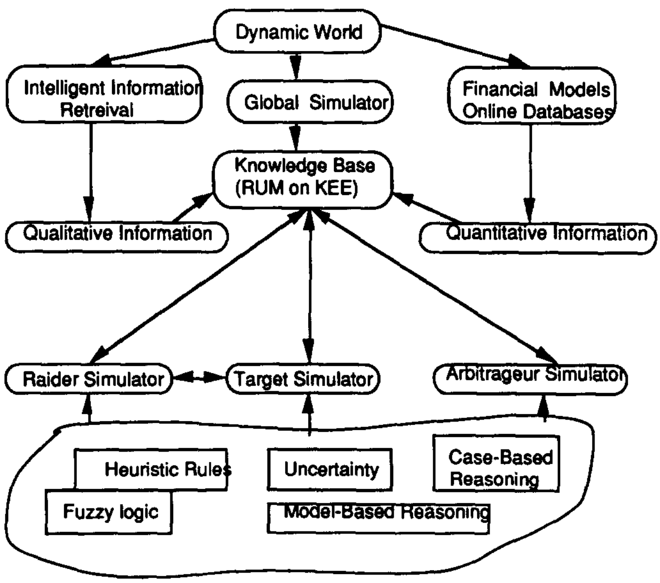

Figure 4. The software architecture of MARS.

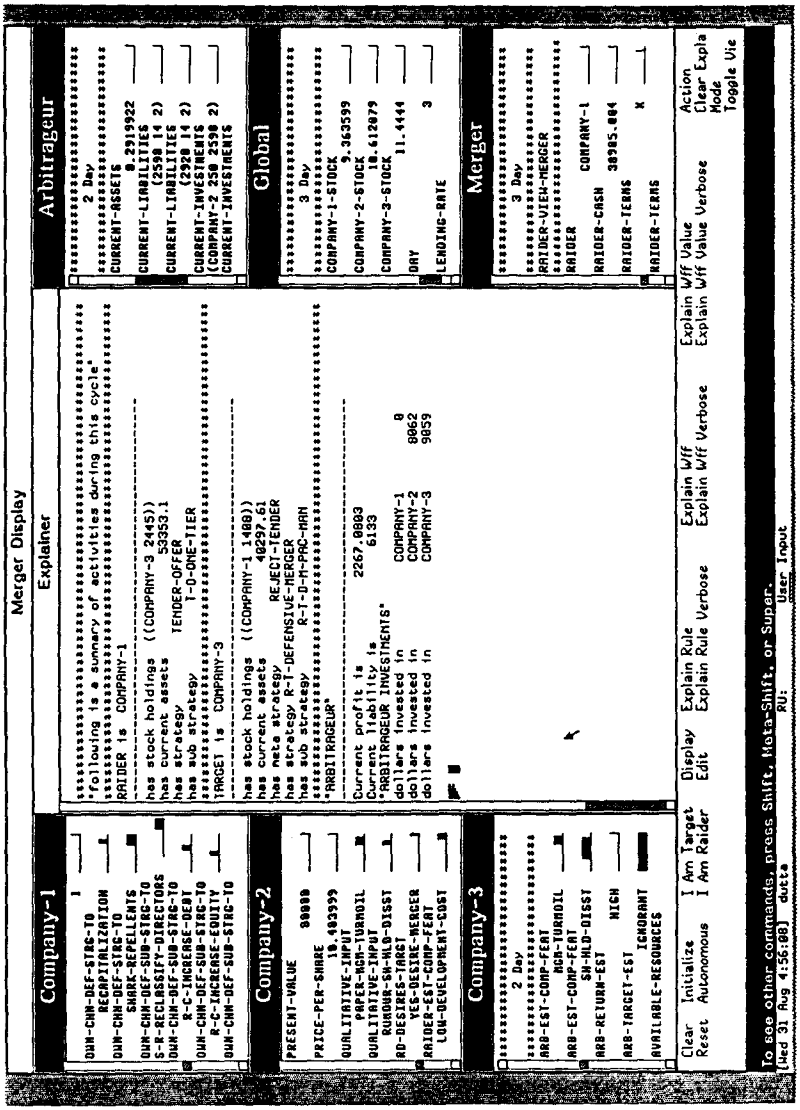

Figure 5. The user interface of MARS.

companies, however, only three of them can be displayed at any one time). The windows to the right in Figure 5 display the values of different attributes of the arbitrageur, global environment, and the merger deal currently in progress. The central window contains a summary of the activities during the last execution cycle of MARS. More details on the structure, implementation, and use of MARS can be found in [17].

4.2 RBR in MARS

MARS is implemented using RUM/PRIMO [41, 42] and KEE. Though RUM/PRIMO uses the data structures and graphical interface of KEE, it has its own rule system. We describe below how some of the important issues in RBR (mentioned in Section 1.2.1) are addressed in MARS. Parts of Subsections 4.2.1 and 4.2.2 review certain aspects of uncertainty management within RUM/PRIMO necessary for understanding the integration of RBR and CBR explained in the following parts of the paper. The description is tutorial in nature and is included for completeness. Readers familiar with references [41, 42] may choose to skim over it.

4.2.1 NATURE OF FACTS AND RULES Facts are qualified by a degree of *confirmation* and *refutation*. For a fact A, the lower bounds of the degrees of confirmation and refutation are denoted by $L(A)$ where $L(\bar{A})$, respectively. The following identity holds:

$$L(\bar{A}) = 1 - U(A)$$

where $U(A)$ denotes the upper bound of the degree of confirmation in A. Note that $L(A) + L(\bar{A})$ need not necessarily be equal to 1, as there may be some ignorance about A which is given by $(1 - L(A) - L(\bar{A}))$. The degree of confirmation and refutation for the proposition A can be written as the interval $[L(A), U(A)]$. These weights are viewed as compositional truth values in a multiple-valued logic. Figure 6 contains a screen dump from the MARS system specifying the different values of the different attributes used to specify a company. Each value has an associated pair $[L(A), U(A)]$, represented graphically to the right of the individual values. The black band delimits the ignorance about that particular value and the white band to the left(right) of the black band specifies the degree of confirmation(refutation) of that value.

RUM/PRIMO provides a natural representation for plausible rules. Rules are discounted by *sufficiency* (S), indicating the strength with which the antecedent implies the consequent and *necessity* (N), indicating the degree to which a failed antecedent implies a negated consequent. Note that conventional strict implication rules are special cases of plausible

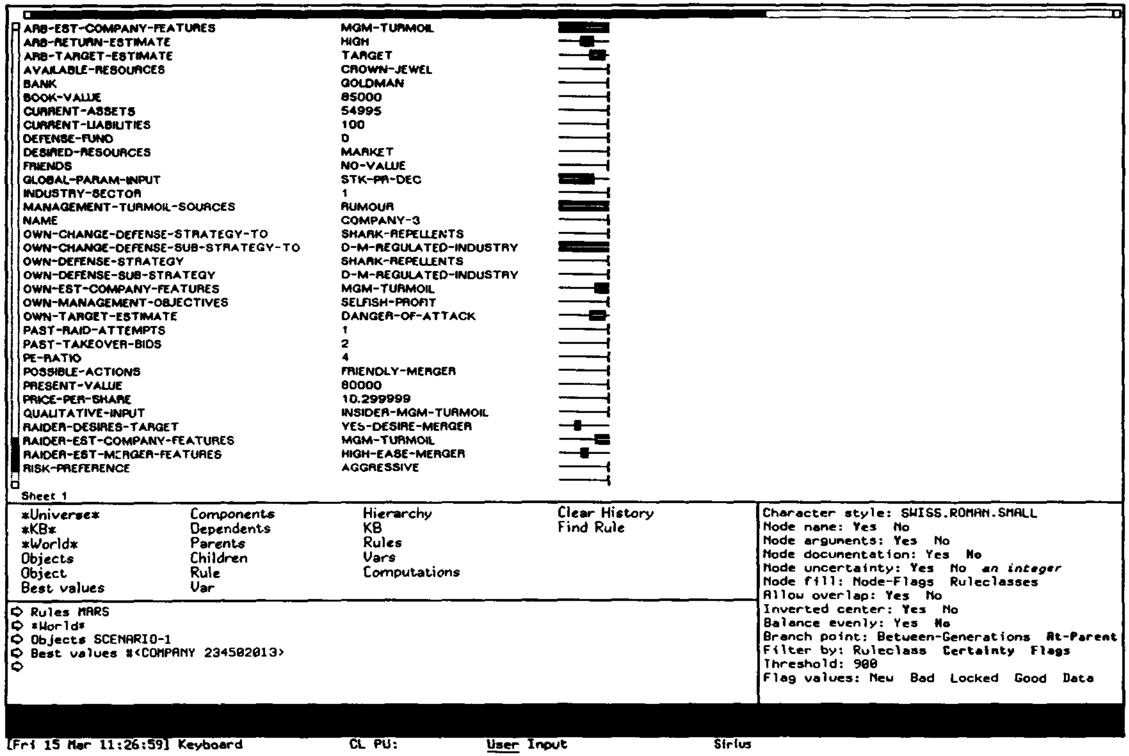

Figure 6. Uncertainty in the values of different attributes within MARS.

rules with $S = 1$ and $N = 0$. Each rule has an associated *context* that represents the set of preconditions determining the rule's applicability to a given situation. It is the responsibility of the designer of the knowledge base to determine the appropriate contexts for rules, just as he/she determines the various premises and conclusions of rules. Rules in the MARS knowledge base are clustered into a hierarchical collection of rule classes (see Section 4.2.3). Typically, all rules within a certain rule class tend to have the same context. Thus contexts also serve as a useful mechanism for guiding the inference process down a particular path in the knowledge hierarchy. An example of a rule and its associated context is provided in the following paragraphs.

The general structure of a RUM/PRIMO rule is as follows:

```
(Rule-Name Knowledge-Base-Name
 Premise-List
 Consequence-List
 Premise-Slots
 Context-List                  ;Optional
 Sufficiency-Necessity-Pair    ;Optional
 T-Norm-Operator               ;Optional
 Context-Threshold             ;Optional
 Context-Flag                  ;Optional
 Rule-Daemon)                  ;Optional
```

As shown, a rule consists of certain mandatory parts such as a *rule-name*, a *knowledge-base-name* (to which the rule belongs) and lists of premises, consequences, and slots (of the frames or KEE units) used in the premise list. Other components of a rule are optional and default to certain values when not specified. These various components are briefly described below. A knowledge base can also contain rule templates, which when *instantiated* yield rules. For illustration purposes, we reproduce below a slightly edited rule template for the raider from MARS (from the knowledge base company-*kb*):

```
(add-template 'rd-target-mgm-friendly 'company-kb                 ;;line 1
   '((is-value? ?target 'raider-est-company-features
      :friendly-mgm))                                             ;;line 2
   '(('bear-hug 'raider-desires-strategy :yes-desire-strategy))   ;;line 3
   '((strategies-not applied '(bear-hug) current-day))            ;;line 4
   '(*very-high-chance* *high-chance*))                           ;;line 5
   t3                                                             ;;line 6
   '(b-h-selection                                                ;;line 7
   900)                                                           ;;line 8
```

This rule template called *rd-target-mgm-friendly* from the knowledge base *company-kb*, when instantiated for a given world state (i.e., a given raider and target) produces a rule in MARS. The (simplified) rule states that if the raider estimates that the target management is friendly (*premise—line 2*), then there is a very high chance (*sufficiency—line 5*) that he/she desires the bear-hug attack strategy (*conclusion—line 2*). Otherwise, there is a high chance (*necessity—line 5*) that he/she will not desire such a strategy. This rule is to be activated only if the bear-hug strategy has not been already applied (*context—line 4*). This context must be confirmed to a degree higher than 900/1000 (*threshold—line 8*) for it to be valid.

Line 6 give the T-Norm operator used to aggregate the uncertainty in the rule and line 7 gives the name of the rule-class of which this particular rule is a member. A more detailed description of the meaning and implications of lines 5 and 6 is given below. Line 7 gives the name of the *rule-class* (see Section 4.2.3) in the *company-kb* knowledge base to which this rule belongs. Fuzzy logic [25] can be used in the rules to specify values in the above rule. For example, the sufficiency and necessity measures (**very-high-chance** & **high-chance*—line 5*) are fuzzy sets, with trapezoidal distribution functions. The use of fuzzy logic enables a much richer representation of inprecision in rule premises, conclusions, and contexts.

4.2.2 INFERENCE IN RULE GRAPH RUM/PRIMO provides an uncertainty calculus based on a set of five Triangular norms (T-norms) [32–35] for inference in the rule graph. *T-norms* and *T-conorms* are two-place functions from $[0,1] \times [0,1]$ to $[0,1]$ that are monotonic, commutative, and associative. They are the most general families of binary functions that satisfy the requirements of the conjunction and disjunction operators, respectively. Their corresponding boundary conditions satisfy the truth tables of the logical AND and OR operators. Five uncertainty calculi based on the following five *T-norms* are defined in RUM/PRIMO:

$$
\begin{aligned}
T_1(a,b) &= \max(0, a+b-1) \\
T_{1.5}(a,b) &= (a^{0.5}+b^{0.5}-1)^2 \text{ if } (a^{0.5}+b^{0.5}) >= 1 \\
&= 0 \text{ else} \\
T_2(a,b) &= ab \\
T_{2.5}(a,b) &= (a^{-1}+b^{-1}-1)^{-1} \\
T_3(a,b) &= \min(a,b)
\end{aligned}
$$

Their corresponding DeMorgan dual *T-conorms*, denoted by $S_i(a,b)$, are defined as

$$S_i(a,b) = 1 - T_i(1-a, 1-b).$$

These five calculi provide the user with an ability to choose the desired uncertainty calculus starting from the most conservative (T_1) to the most liberal (T_3). $T_1(T_3)$ is the most conservative (liberal) *T-norm* in the sense that for the same input certainty ranges of facts and rule sufficiency and necessity measures, $T_1(T_3)$ will yield the minimum (maximum) degree of confirmation of the conclusion. For each calculus (represented by the above five *T-norms*), the following four operations have been defined in RUM/PRIMO:

- **Antecedent Evaluation** To determine the aggregated certainty range $[b, B]$ of the n clauses in the antecedent of a rule, when the certainty range of the i^{th} clause is given by $[b_i, B_i]$:

$$[b, B] = [T_i(b_2, b_2, \ldots, b_n), T_i(B_1, B_2, \ldots, B_n)].$$

- **Conclusion Detachment** To determine the certainty range, $[c, C]$ of the conclusion of a rule, given the aggregated certainty range, $[b, B]$ of the rule premise and the rule sufficiency, s, and rule necessity, n:

$$[c, C] = [T_i(s, b), 1 - (T_i(n, (1 - B)))]$$

- **Conclusion Aggregation** To determine the consolidated certainty range $[d, D]$, of a conclusion when it is supported by m ($m > 1$) paths in the rule deduction graph, that is, by m rule instances, each with the same conclusion aggregation *T-conorm* operator. If $[c_i, C_i]$ represents the certainty range of the same conclusion inferred by the i^{th} proof path (rule instance), then

$$[d, D] = [S_i(c_1, c_2, \ldots, c_m), S_i(C_1, C_2, \ldots, C_m)].$$

- **Source Consensus** To determine the certainty range, $[L_{tot}(A), U_{tot}(A)]$ of the same evidence, A, obtained by fusing the certainty ranges, $[L_i(A), U_i(A)]$, of the i^{th} information source out of a total of n different possible information sources:

$$[L_{tot}(A), U_{tot}(A)] = [\text{Max}_{i=1, \ldots, n} L_i(A), \text{Min}_{i=1, \ldots, n} U_i(A)].$$

Line 6 in the sample rule template described earlier specifies the *T-norm* (T_i) operator selected to propagate uncertainty in the rule using the operations defined above. The use of these operations for inference in a simple rule graph is shown diagrammatically in Figure 7. The certainty ranges at various stages are depicted graphically. Note that the *Conclusion Aggregation* operation aggregates proof paths (rules) using the same *T-norm*

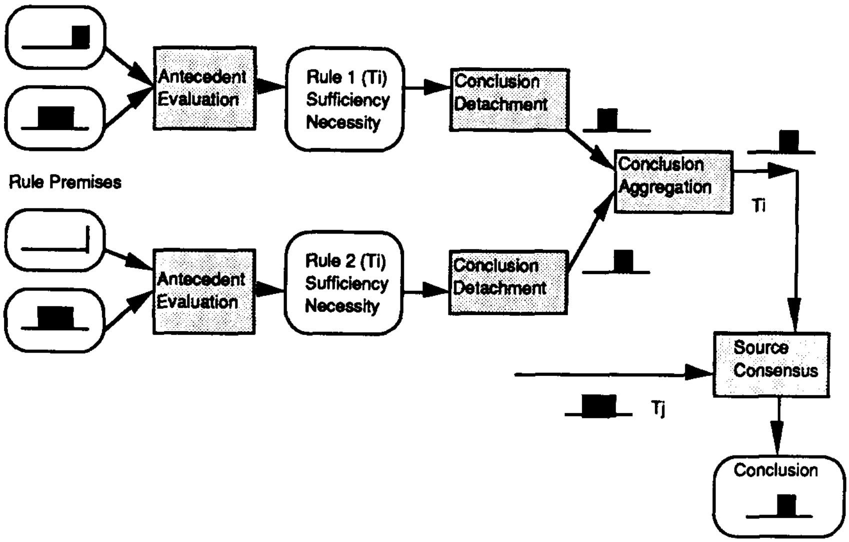

Figure 7. Inference using uncertainty in rule graph.

operator (T_i), whereas the *Source Consensus* operation aggregates proof paths using different *T-norms* (T_i and T_j) operators.

To observe how MARS uses the uncertainty calculi described above, consider the attribute "OWN-TARGET-ESTIMATE" shown in Figure 6. The value "DANGER-OF-ATTACK" of this attribute specifies the degree of confirmation of a company that it will be subject to a takeover attempt in the near future (required for planning necessary defensive strategies prior to an actual takeover attempt). The graphical bar-like representation to the right of the phrase "DANGER-OF-ATTACK" specifies the degrees of confirmation and refutation of this value. Figure 8 contains a screen-dump from the MARS system depicting the *conclusion-aggregation* and *source-consensus* operations used to aggregate the multiple proof paths contributing to the final degrees of confirmation and refutation of the value "DANGER-OF-ATTACK". (Note the similarity between Figures 7 and 8). Each narrow rectangular box to the left of Figure 8 represents one proof path contributing to the determination of the value "DANGER-OF-ATTACK." The consolidated degrees of confirmation and refutation provided of each proof path is shown inside the corresponding boxes. Note that different rule sets (proof paths), in turn, contribute to the consolidated degrees of confirmation and refutation of each of these proof paths. By clicking on the box containing each proof path, it is possible to observe how the four different uncertainty calculi operations (described above)

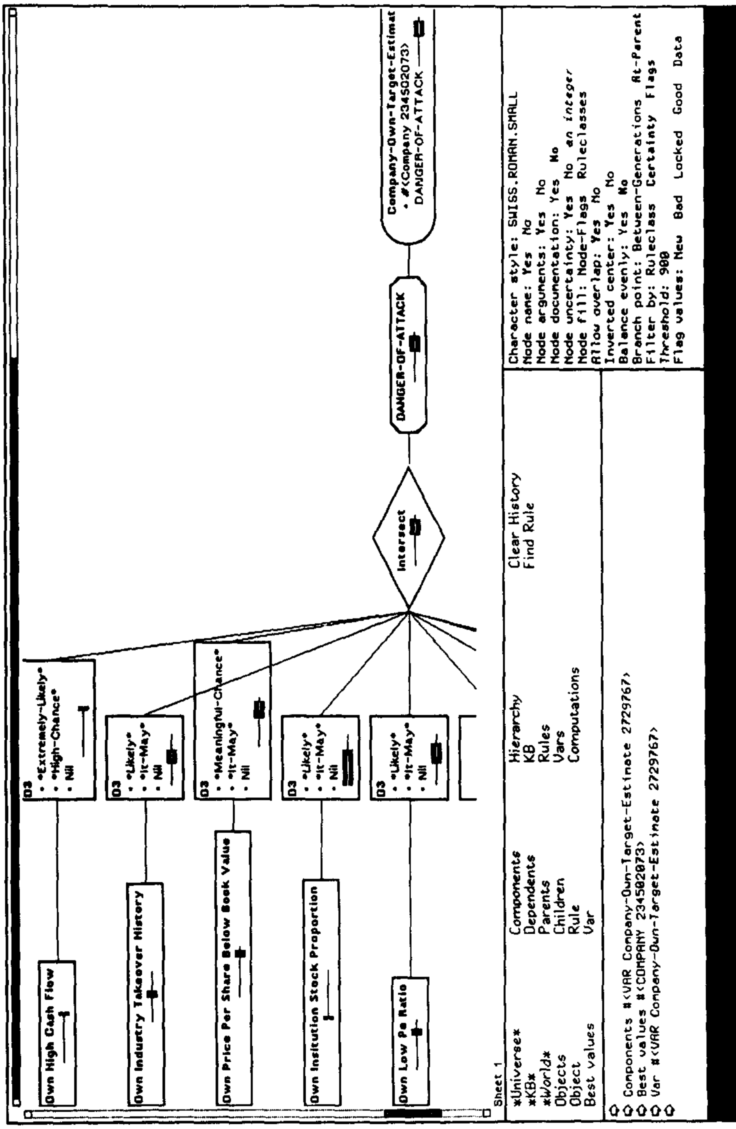

Figure 8. Screen dump of interference process in MARS.

were used to obtain the consolidated degrees of confirmation and refutation of the corresponding proof path. It is thus possible to trace the inference process in either the forward or backward directions. Of particular interest in Figure 8 is the proof path labelled "Own Industry Takeover History." Evaluation of this proof path causes the CBR module to be triggered. The bar-like representation contained with this box represents the final contribution of CBR to the value "DANGER-OF-ATTACK" of the attribute "OWN-TARGET-ESTIMATE" for a certain company. Thus Figure 8 depicts the final stage in the combination of CBR with RBR in MARS. More details on this integration are provided in Section 4.

The theory of RUM/PRIMO is anchored on the semantics of many-valued logics. Unlike other probabilistic systems, RUM/PRIMO's reasoning mechanism is based on a truth functional, many-valued logic anchored on the notion of similarity. References [32, 33] describe a comparison of RUM/PRIMO with other reasoning with uncertainty systems, such as Modified Bayesian [43], Certainty Factors [1, 44], Dempster–Shafer [45, 46], and Fuzzy logic [25].

4.2.3 STRUCTURE OF KNOWLEDGE BASE The knowledge base of MARS contains approximately 550 rule-templates. Though not a very large number, it is sufficiently large to warrant structuring the knowledge base due to reasons explained in Section 1.2. Related rules are clustered into subsets called *rule-classes* and these rule classes are organized into an abstraction hierarchy with specialization as one traverses down the hierarchy. Figure 9 depicts a partial abstraction hierarchy of rule classes from the MARS knowledge base for the selection of raider strategies. Note that the rule class titled RAIDER-EST-ANTI-TRUST-SUCCESS-RULES contains rules used by the raider to determine the possibility that an anti-trust move will succeed. The rules in this rule class are further divided into two categories—rules related to the effect of prior cases (under RD-ANTI-TRUST-PREC) and those related to other issues governing the success of the anti-trust move (under RD-ANTI-TRUST-WILL-SUCCEED). This common representation of rules related to both prior cases and domain-specific issues facilitates the seamless integration of RBR and CBR in MARS. The next few sections provide more details on this integration. The contexts of the rules in the rule-class collectively help define the conditions under which that particular rule-class is activated. Thus during inference, it is possible to focus and selectively search the rule graph so as to reach the desired solution as efficiently as possible.

The expressiveness of RUM/PRIMO is enhanced by another function: *belief revision*, which is essential to the dynamic aspect of the domain of M & A. The belief revision mechanism detects changes in the input, keeps track of the dependency of the intermediate and final conclusions of these

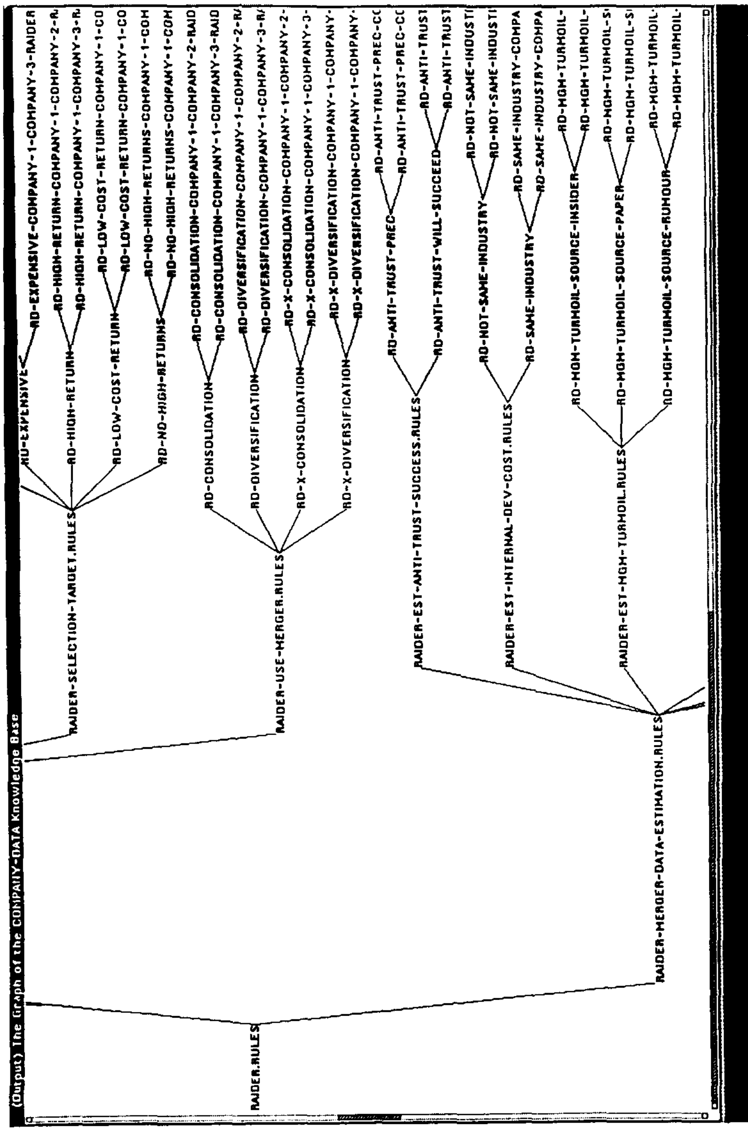

Figure 9. Partial structure of knowledge base of MARS.

inputs, and maintains the validity of these inferences. For any conclusion made by a rule, the mechanism monitors the changes in the certainty measures that constitute the conclusion's support. As a simple example, consider the inference process depicted in Figure 7. Assume that the conclusion has been computed given some particular values of the input premises of rules 1 and 2. The depicted proof path is stored by the inference engine of RUM. Each fact (node in the proof path) has an associated flag. These flags can represent various states, but for simplicity consider only two: *good*, *bad*. Initially the flags of all nodes in the inference path of Figure 7 are "good." As the environnment is dynamic, the flags of various nodes can change. Assume that one of the input premises of rule 1 changes (causing its flag to change from "good" to "bad"). Now the belief revision mechanism of RUM/PRIMO would automatically change the flags of all dependent nodes in the inference graph to "bad," that is, the flags of Rule-1 conclusion and the aggregated conclusion would be changed to "bad" (the flag of rule-2 conclusion would still be "good"). Later, if the values of rule-1 conclusion or the aggregated conclusion are required, then the inference engine of RUM/PRIMO would notice that their flags are "bad" (indicating that things have changed since they were last computed) and recompute the values (nodes in the proof path) whose flags are "bad." Note that this recomputation is "lazy," that is, the new values are only computed when required and not when the flags are changed (to "bad").

4.3 CBR in MARS

We will now turn our attention to the CBR component in MARS. We have outlined earlier (see Section 1.2) the important issues within CBR. We explain below how each of these issues has been addressed in MARS.

REPRESENTATION Given our intention to integrate RBR and CBR within the common architecture of rules, we decided to represent individual cases in the MARS case library as RUM/PRIMO rule templates. The general structure of a rule template has been described in Section 4.2.1. To see how cases are represented as rule templates, let us refer to Section 3.3: the part of the Mobil–Marathon case described there is represented as the following (edited) rule template (expressed in pseudo English & Lisp):

(CASE 1)

IF (*similar-industry ?raider ?target*) *AND* (T_i)
(*large-merged-national-market ?raider ?target*) *AND* (T_i)
(*significant-local-dominance ?raider ?target*)
THEN 0.9 (*sufficiency*)
(*anti-trust-success ?raider ?target*)

where (T_i) is the *T-norm* operator chosen for the conjunction of the rule premises. This rule template expresses the following: The facts that Mobil and Marathon operated in a similar industry sector and that the combined entity (if Mobil succeeded in taking over Marathon) would have a large national market share and significant local dominance were very important (to the degree 0.9) in determining the success of the anti-trust move. Each premise (*similar-industry*, *large-merged-national-market* and *significant-local-dominance*) is an *abstract feature* and is implemented as a call to a procedure that returns an interval-valued certainty measure (see Sections 4.2.1 and 4.2.2) when the variables *?raider* and *?target* have been instantiated to a particular raider and target. This interval value reflects the degree of match of the case with the probe—details are provided in later sections. The sufficiency measure, 0.9, gives the degree to which the conjunction of the three premises is relevant for determining the success of the anti-trust suit in this case. The necessity measure has been omitted for clarity.) It should be noted that Mobil and Marathon have been replaced by the role variables *?raider* and *?target*, respectively. The uncertainty mechanism supported by RUM/PRIMO rules is being used in this representation for two purposes: first, to represent the relative importance of the premises for the conclusion (by the choice of T_i—see Section 4.2.2), and second, to represent the relevance of the premises to the conclusion (via the *sufficiency* and *necessity* measures).

The above rule template gives an *interpreted* description of a particular aspect of the Mobil–Marathon case, that is, it states explicitly what the reasons (and their relative degrees of importance) were for the success of the anti-trust move in the Mobil–Marathon case rather than merely stating that the anti-trust move succeeded. It is important to consider means to obtain the interpreted rule templates from available data. This is possible in the domain under consideration. Consider the ruling of Judge Manos in the Mobil–Marathon case which outlined detailed reasons for the judgment. The above rule template is merely a translation of Judge Manos' ruling. The case rule templates in MARS are currently inputted by a manual reading of prior rulings regarding the legal/tax aspects of the M & A domain. It is, of course, interesting to consider automating this process of extraction of case rule templates from case descriptions. One possibility is to use an intelligent information retrieval system to read and "understand" such legal rulings and extract the reasons for certain actions/ decisions. This is not very difficult to do with rulings as given by Judge Manos because the statements in the ruling themselves explicitly state the rationale behind the described actions/decisions. SCISOR [47] is an intelligent natural language system that can possibly perform this function. Our future plans include looking into the use of SCISOR for such purposes. Automating the extraction of case rule templates in a general domain is a more difficult problem. The process employed will be intimately related to the schema used for representing prior case descriptions. If the case description consists only of uninterpreted descriptions of what happened previously, then a significant amount of domain knowledge will be required to find out "why certain events happened" or "what is the rationale behind the decisions made." This shall entail the construction of a large knowledge-based system only for the purposes of extraction of case rule templates. This problem is obviated during the construction of the case library of MARS as prior legal rulings usually provide explicit reasons and rationales for decisions.

Our choice of rule templates as the stored representation of cases in the case library departs from the conventional representation (frames and MOPS [5]) of cases (Section 1.2). We feel that this departure is justified as it is possible to achieve a seamless integration of CBR and RBR in MARS using such a representation. This would have been very difficult had we used different representations for cases and rules. Also, as RUM/PRIMO rules are richer data structures (with *contexts*, *thresholds*, *sufficiency & necessity measures* and *fuzzy values*) than simple categorical IF... THEN... rules, it is possible to adequately represent many features of cases required for CBR. One disadvantage of using rule templates to represent cases is that they cannot be used if some of the rule premise (abstract feature) data is not available for the current problem (probe).

STRUCTURE OF CASE LIBRARY Any particular case, such as the Mobil–Marathon case described earlier, has many other interesting aspects (besides the anti-trust suit), and there are many other cases in the case library that deal with the topic of anti-trust moves (e.g., cases related to *Brown Shoe*, *Kinney Co.*, and *Pabst Brewing Company*). In the MARS case library, a *hierarchical*, *functional* structure is imposed on the cases in the case library. Figure 10 illustrates the partial structure of the case library.

As shown, the case library has at the top level two branches, one containing cases pertaining to *defensive* strategies and the other related to *attack* strategies. Within the defensive strategies category, we have subcategories for cases related to different types of defensive strategies (e.g., *pac-man*, *greenmail*, and *anti-trust*). Going further down the subcategory for *anti-trust* defensive strategy cases, we have sub-subcategories for cases related to *market dominance*, *efficiency*, and *foreign competition*. The example rule template, CASE 1, (described above) is contained in the market dominance category. Note that all case descriptions relevant for the *market-dominance* aspect of the *anti-trust* move are grouped together. Thus the templates for Pabst Brewing Company and Brown Shoe–Kinney all appear together with the template for Mobil–Marathon. Note also that all aspects of a particular case are not stored together. Thus templates

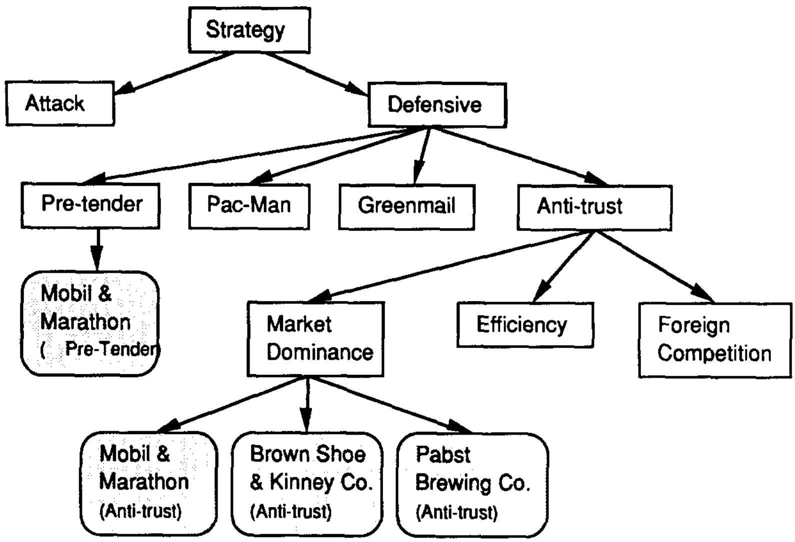

Figure 10. Partial structure of case library in MARS.

pertaining to different aspects of the Mobil–Marathon case are distributed appropriately in different parts of the case library hierarchy. For example, the rule templates expressing the *pre-tender* defensive strategies related to the Mobil–Marathon case are stored under the subcategory *pre-tender* defensive strategies as shown in Figure 10. For simplicity, the different types of pre-tender defensive strategies (e.g., *poison pills*, *shark repellants*, and *crown jewel lock ups*) are not shown in Figure 10.

As mentioned earlier, flat memory structures and discrimination networks are popular methods for structuring the case library in the literature. Flat structures are expensive to search sequentially and thus discrimination networks are more widely used. The hierarchical, functional structure of the MARS case library, although similar in many aspects to a discrimination network, is different in emphasizing the functional breakup of a case into different parts of the case library and grouping of similar parts of many different cases into the same part of the case library. Although such an approach has some disadvantages (as discussed in Section 5.2), it forces a cleaner structure on the case library and yields simpler retrieval mechanisms.

NEED FOR CBR It is important to recognize two points. First, CBR is important only for certain problems and goals; it is not useful to always consider CBR. For example, in the domain of M & A, CBR is useful primarily for structuring the legal and tax aspects of the deal. For some other aspects, such as the use of statistical financial models, it makes little sense to include CBR. Second, CBR is, in general, only one approach (proof path) to the solution of a goal/problem. There are (usually) other approaches (or proof paths) to the same goal/problem, and it is important to consider the contributions of all paths, proportional to their relative importance. These aspects are significant, as most research in CBR has considered it in isolation till now.

In MARS, the inference process can be considered as the traversal of paths in a rule graph. Premises, qualified by certainty intervals, combine (using RUM's uncertainty calculi) to generate conclusions (also qualified by certainty intervals) that either act again as premises for other rules or generate final conclusions. The decision whether to activate CBR or not is left to the designer of the system. Whenever the expert (or system designer) feels that CBR is important for deciding about a particular conclusion, a rule to this effect is added in the knowledge base. For example, consider a hypothetical M & A deal, $M1$, being analyzed by MARS. As precedents are important for the evaluation of the possibility of success of an anti-trust suit in $M1$, a rule to this effect is added (under the rule class *RD-ANTI-TRUST-PREC* [see Figure 8]):

(RULE 1)

IF *similar anti-trust precedent exists*
THEN 0.8 (*sufficiency*)
anti-trust successful in $M1$.

This rule (shown as rule 1 in Figure 11) states that *if there is a precedent of an anti-trust move succeeding in a similar situation* (*as in* $M1$), *then that might cause the anti-trust move to succeed in* $M1$ *to the degree of* 0.8. Rule 1 shown above is, of course, a simplified English-like representation of an actual rule and omits features like *context lists*, *thresholds*, and the *necessity* measure; however, the presence of a prior precedent is not the only fact affecting the outcome of an anti-trust move in $M1$. There are, in general,

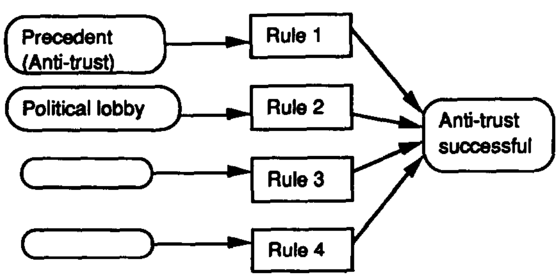

Figure 11. Example of a simple rule graph.

other factors (e.g., the "*political lobbying power of the target*") that also either confirm/disconfirm the conclusion. A rule (shown as RULE-2 in Figure 11) is, therefore, added (under the rule class *RD-ANTI-TRUST-WILL-SUCCEED* [see Figure 9]):

(RULE 2)

IF *target has strong political lobby*
THEN *it is likely* (*sufficiency*)
*anti-trust successful in M*1.

The above two rules represent two different proof paths, each contributing to the determination of the goal "*anti-trust successful in M*1" (see Figure 11). The *conclusion aggregation* and *source consensus* operations (see Section 4.2.2 and Figure 7) determine the relative contributions of RULE-1 and RULE-2 to the final conclusion of "*anti-trust successful in M*1."

Note that the uncertainty mechanism of RUM/PRIMO is being used to (1) represent the importance of a prior precedent for the current conclusion (RULE-1) and (2) aggregate the relative importance of different proof paths—of which CBR is but one possible proof path.

This emphasis on *explicitly recognizing the need for CBR* is novel to our research. As mentioned before, most prior research in CBR has ignored this issue, having considered CBR in isolation. The work by Rissland and Skalak [2] has also studied the need for deciding when to use CBR; however their approach is very different as it requires a central scheduler to decide upon the invocation of CBR based on some heuristics. They do not consider the use of CBR and RBR as parallel proof paths. They use either CBR *or* RBR at any one time, depending upon which is feasible. This is in contrast to our approach in which both CBR and RBR are considered as two equal proof paths in parallel, and each makes a proportional contribution to the confirmation or disconfirmation of the conclusion.

RETRIEVAL AND SIMILARITY METRICS Recall that a RUM/PRIMO rule template (Section 4.2.1) has a context that keeps track of the environment in which that rule is activated. This context of rules is used to efficiently index into the hierarchical structure imposed on the case library. For example, if the context indicates that anti-trust success is being evaluated, only relevant cases will be retrieved by following down the appropriate path in the case library hierarchy (see Figure 10).

In conventional CBR, the general approach is to retrieve *one* prior case that is most *similar* to and *relevant* for the current goal/problem. The approach taken in MARS departs from this in that it retrieves all *relevant* cases and then proportionately weighs the relative contribution of each relevant case. For an example, consider Figures 10 and 11. In Figure 10, there are three *case rule templates* under the *anti-trust* branch. When the inference engine of MARS encounters RULE-1 (in Figure 11), it goes into the case library hierarchy (Figure 10) and retrieves all relevant cases (here the three *case rule templates*: *Mobil–Marathon*, *Brown-Shoe-Kinney* & *Pabst-Brewing*).

The matching process is operationalized by instantiating the case rule templates in the case library to the situation of the current world, $M1$. This process converts the case rule templates to rules that can be used in the reasoning process of $M1$ and at the same time determines the degree of relevance of the previous cases to $M1$. Thus if CASE-1 (case rule template) were instantiated to $M1$ world conditions, the variables *?raider* and *?target* would be instantiated to the raider and target, respectively, in $M1$, and each of the three premises will be evaluated to yield certainty ranges that give the degree to which the premises of the case are true in the current $M1$ world. If they are not relevant (true), then a very low confirmation for the premises will be obtained and vice versa. Using the uncertainty calculi of RUM/PRIMO (Section 4.2.2), CASE-1 will yield a conclusion with a certainty range which is the degree to which that case similar to and relevant for $M1$. As there will be many cases for the same conclusion (e.g., successful anti-trust cases) in the case library, an aggregated value of the relevance of all the previous cases can be obtained using the *conclusion aggregation* and *source consensus* operations of RUM's uncertainty calculi. The node labelled "*anti-trust success*" represents the aggregated contribution of various cases for determining the success of an anti-trust suit in $M1$.

The interval-valued certainty ranges obtained when the premises of case rule templates are instantiated to the current world conditions are the same as the intervals, $[N(p|d), P(p|d)]$, representing the lower and upper bounds on the degree of match between the pattern (cases) and data (probes). Here the *necessity measure* $N(p|d)$ represents the degree of semantic entailment of a pattern descriptor p given a datum d, and the *possibility measure* $P(p|d)$ represents the degree of intersection between the same pattern and datum.

ADAPTATION Once the relevant cases have been retrieved from the case library, the *adaptation* phase of CBR in MARS is performed by the process of *reinstantiation* of the case rule templates to the current world situation and the application of the uncertainty propagation mechanism of RUM/PRIMO (Section 4.2.2).

The process of *reinstantiation* is closely intertwined with the matching process as described above in the section on *Retrieval & Similarity Metrics*. When CASE 1 (case rule template) is reinstantiated to the current situation ($M1$), the certainty measures associated with the evaluation of each of the premises indicates the degree to which those premises are valid in the current situation, or in other words, it represents the result of the *adaptation* of the premises to the current problem. The aggregated uncertainty of the premises is propagated forward using the uncertainty calculus supported by RUM/PRIMO to contribute to the confirmation/disconfirmation of the conclusion (node labelled Precedent (anti-trust) in Figure 12); however, the process of adaptation is not complete at this stage as there are other relevant cases that also have to be considered. The other case rule templates (*Brown-Shoe-Kinney* and *Pabst-Brewing* in Figure 10) will also yield on reinstantiation similar contributions to the confirmation/ disconfirmation of the conclusion (node labelled *precedent* [*anti-trust*] in Figure 12). Using the *conclusion aggregation* and *source consensus* operations (Section 4.2.2), the proportional contributions of all three case rule templates can be combined to yield an aggregated measure of confirmation/ disconfirmation for the conclusion "*precedent* (*anti-trust*)." This process has been illustrated in Figure 12.

The next stage in the adaptation phase is to notice that the nodes labelled *precedent* (*anti-trust*) are the same in Figures 11 and 12. Once the aggregated contribution of all relevant cases has been obtained (by the process described above), it gives the degree of confirmation/ disconfirmation of the premise of RULE-1 (Figure 11). The net contribution of CBR to the current conclusion—"*anti-trust successful* in $M1$"—(node labelled *anti-trust successful* in Figure 11) can then be obtained by applying the usual *T-norm* calculus to RULE-1.

The adaptation procedure described above is a variation of *parametric adaptation* when the goal variable (conclusion—node labelled *Precedent* [*Anti-trust*] in Figure 12) has only one value (anti-trust success in Figure

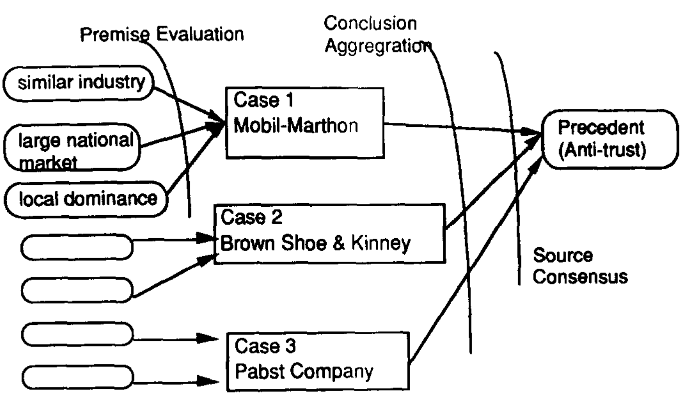

Figure 12. Expansion of case library during case adaptation.

12). In general, the goal variable can have multiple values. For example, the goal variable can be *value of tender offer*, which can take a number of different values in different cases as shown in Figure 13. Each of these values is qualified by a lower and upper bound on its certainty (the thick portions of the lines gives the bounds—using the same graphical notation as in Figure 6) and together they define a distribution. The value given by case-based reasoning has to be obtained by interpolation on such a distribution. The interpolation mechanism chosen is domain dependent.

The adaptation phase of CBR in RUM/PRIMO is also different from that common in the literature. Though *derivational* in approach, it relies extensively on the uncertainty calculus supported by RUM/PRIMO. Indeed, it is the rich *T-Norm* calculus supported by RUM/PRIMO that makes this possible. The above process of adaptation has its limitations in that it does not allow any structural changes (in the case rule templates), but, more important, it allows the adaptation phase to be integrated into the normal inference cycle of RBR in MARS. This was a crucial goal for our research. It also allows the simultaneous consideration of many different relevant cases. By suitably changing the *T-norm* operators, it is possible to easily adjust the relative contributions of various cases in the CBR process. The lack of an ability to make structural changes in the case rule templates can be overcome partly by introducing additional proof paths (e.g., rules 2 and 3 in Figure 11) and proportionately weighing their relative contributions. The belief revision mechanism of RUM/PRIMO (see Section 4.2.3) also monitors changes in the inference graph corresponding to the case rule templates. Thus if facts change, cases can be reevaluated.

INTEGRATION Though most prior research in CBR has considered it in isolation, an important goal of our research was to try and integrate it

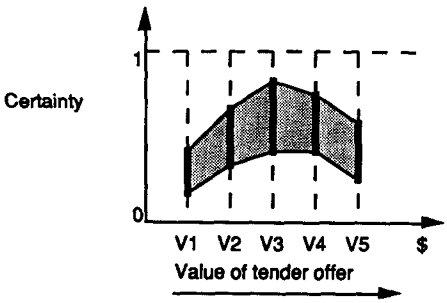

Figure 13. Interpolation with multiple values.

seamlessly into RBR. The motivations for this were explained in Section 1 (also see Figure 3).

The above subsections taken together have outlined the details of the integration of RBR and CBR in MARS. The inference engine of RBR in MARS is the dominant inference cycle and under normal circumstances goes about its task of searching through the rule graph of the MARS knowledge base. When it comes across a rule premise involving CBR (see RULE-1), it begins evaluating the premise much as it does for any other normal (not involving CBR) rule; however, for evaluating the premise of a rule using CBR, it has to go and retrieve the relevant cases from the case library, which results in the expansion and search of an additional rule graph (involving the case rule templates—see Figure 12). Once the rule graph involving the case rule templates (Figure 12; also see Figure 8) has been evaluated, it returns with the confirmation/disconfirmation of the premise of the rule involving CBR (RULE-1) and carries on with the normal inference propagation mechanisms. Note that the only difference in the evaluation of the premise of a rule involving CBR occurs in the fact that although most premises of normal rules require simple matches or procedure calls, the premises of rules involving CBR require the expansion of and search for an additional rule graph (corresponding to the case rule templates).

Thus to summarize the process briefly:

- Cases are stored as rule templates (CASE 1).
- If CBR is important, a rule to this effect (RULE 1) is added.
- Case rule templates are instantiated automatically while evaluating the premise of rules like (RULE 1). Rule contexts are used for indexing into the hierarchical structure of the case library.
- Adaptation of cases is performed by the process of instantiating the case rule templates and using the *T-Norm* calculus to propagate the inference forward.

Finally, we would like to emphasize two points. First, note the seamless and compact integration of RBR and CBR. No changes need to be made in the inference engine of RBR in MARS (which remains the same whether CBR is used or not) for accommodating CBR. Second, the uncertainty calculus supported by RUM/PRIMO plays a major role in the entire process, right from representation up to adaptation and integration.

5. CONCLUSION

This section compares our work with prior research and concludes the paper.

5.1 Comparison with Related Research

Broadly speaking, there are three different possibilities to combining RBR and CBR: (1) keep RBR and CBR as two "equal" reasoning modules with a high-level controller (blackboard) deciding when to activate which reasoning module, (2) let the CBR module use the inference capability of rules when needed, and (3) have the RBR module dominate and activate CRB explicitly at specific points in the reasoning process.

The research of Rissland and Skalak [2] is an example of the first approach. Although working in the legal domain of *statutory interpretation*, they have built a system called *CABARET*, whose architecture consists of two co-equal reasoners, one a RBR and the other a CBR, with a separate agenda-based controller. the central controller contains heuristics to direct and interleave the two modes of reasoning and to post and prioritize tasks for each reasoner. In a more recent version of CABARET [15], this type of control is implemented by a blackboard system, GBB.

The second paradigm is exemplified by the work of Bonissone, Blau, and Ayub in the development of the CARS system [48, 49]. The CBR module in CARS is the dominant system and it activates PRIMO [42], a rule-based reasoner that contains plausible rules of abstraction, evaluation, and modification. PRIMO rules are used in case indexing to augment the case and probe representation with a set of abstract features derived from subsets of surface features. Each abstract feature has a value (with fuzzy semantics) and a certainty evaluation qualifying the value assignment. In case retrieval, the rules combine the similarity measures computed across the abstract features of probe and cases and provide a partial ordering on the cases. In case adaptation, rule-based inferences are used to perform aspects of derivational adaptation to achieve retrieved subgoals.

The research presented in this paper is an example of the third approach to combining RBR and CBR. In MARS, rules are used to represent the domain expertise that is required for structuring various parts of the M & A deal or deciding upon the best course of action. Rules are used in forward and backward chaining mode to make selected inferences. CBR is activated by selective rules that state the need for including CBR in the inference path of CBR.

It is useful to compare our research with that of Rissland and Skalak as the RBR component is more dominant in CABARET as compared to the CARS system. The major differences are as follows. First, we have chosen rule templates (as opposed to frames in CABARET) as the representation schema for cases. This has allowed us to use a similar data representation for both RBR and CBR and obviated the need to "patch together" two different data representation schemes. Second, we have required the user or system designer to explicitly recognize the need for CBR for a particular conclusion. This approach requires some participation from the system designer, but does not require a central scheduler or controller. We feel that it may be difficult to always choose the right heuristics for the controller and design it to perform correctly and adequately in different, complex domains. Third, we do not activate only one reasoning methodology (RBR and CBR) at a time (as in CABARET), but rather consider them in parallel as two equal proof paths, each making a proportionate contribution to the final conclusion/goal. Fourth, the case library and the retrieval mechanisms in MARS are structured so as to consider the relative contributions of all relevant cases to the conclusion. Finally, our approach to integrating RBR and CBR provides a treatment of uncertainty and approximate matching between input and cases, which is not available in CABARET.

5.2 Contributions and Limitations

We feel that the primary contribution of this paper has been to illustrate the compact and seamless integration of RBR and CBR as implemented in MARS. Both RBR and CBR are very important reasoning methodologies and there has been comparatively little prior work in integrating the two. We hope that this paper represents a significant effort in that direction. Both RBR and CBR are required for solving complex real-world problems. By choosing RBR as the base architecture for integration, we have illustrated a method for adding more power to rule-based systems, i.e., expanding their inference capabilities. Our architecture treats the contributions of CBR and RBR simultaneously and proportionately (according to their relative importance) as separate proof paths to a conclusion. This does not require the use of any special heuristics or agendas. As shown, no changes have to be made to the inference engine of RBR to accommodate CBR. Furthermore, this seamless integration is anchored on common similarity-based possibilistic semantics for both CBR and RBR. The methodology presented in this paper is general and also applies to RBR without uncertainty (where rules and facts are special cases of RUM/PRIMO rules and facts) as well as both problem-solving CBR and precedent-based CBR (as long as the base architecture is rule-based).

We will conclude our discussion by both noting some of the limitations of the methodology described in this paper and proposing future efforts aimed at strengthening this approach. The case library consists of interpreted rule templates (cases) which are conditioned on specific goals. The process of interpretation of data to obtain such rule templates, though possible, is nontrivial. The important goals (to be included in the case library) and the salient features for each goal (expressed in the rule templates representing cases) have to be explicitly engineered into the system. Also, the case library at present has to be necessarily incomplete (as it is not possible to represent all possible relevant features/reasons for all possible events/actions). The system designer has to decide the importance of various goals, the relevancy of different features for the goals, and construct the case library. Any automation of this process would be a significant improvement. Some algorithms for automating the determination of the relevancy of attributes for a goal are given in [50] and we are looking into the incorporation of those techniques into MARS. The hierarchical, functional structure imposed on the case library while making retrieval easy and manageable, does impose restrictions on frequent changes and updates in the structure of the case library. As implemented, we have chosen to retrieve all relevant cases; however, this may not be such a good idea if the number of cases increases to a very large number. Under such conditions, we will have to consider means to limit the number of cases retrieved, by imposing either simple measures like thresholds or more complex measures like dynamic relevancy metrics. All these issues will be the focus of our research goals for the future.

ACKNOWLEDGMENT

The support and help given by General Electric and various members of the AI program at GE Corporate R & D is gratefully acknowledged.

References

1. Buchanan, B. G., and Shortliffe, E. H., *Rule-Based Expert Systems*, Addison-Wesley, Reading, MA, 1984.
2. Rissland, E. L., and Skalak, D. B., Combining case-based and rule-based reasoning: A heuristic approach, in *Proceedings of the Eleventh Joint Conference on Artificial Intelligence*, 524–530, Morgan Kaufmann, San Mateo, CA, August 1989.
3. Barletta, R., and Mark, W., Explanation-based indexing of cases, in *Proceedings of the National Conference on Artificial Intelligence*, 541–546, Morgan Kaufmann, San Mateo, CA, 1988.
4. Ashley, K., and Rissland, E., A case-based approach to modelling legal expertise, *IEEE Expert* 70–77, Fall 1988.
5. Riesbeck, C. K., and Schank, R. C., *Inside Case-Based Reasoning*, Lawrence Erlbaum Associates Inc., NJ, 1989.
6. Schank, R., Towards a general content theory of indices, in *Proc. AAAI Spring Symposium on Case Based Reasoning*, 36–40, 1990.

7. Carbonell, J., and Velose, M., Integrating derivational analogy into a general problem solving architecture, in *Proceedings of the Case-Based Reasoning Workshop*, 104–124, Morgan Kaufmann, San Mateo, CA, May 1988.

8. Velose, M., and Carbonell, J., Integrating analogy into a general problem solving architecture, in *Intelligent Systems: State of the Art and Future Directions*, (Z. W. Ras and M. Zemankova, Eds.), 29–51, Ellis Horwood, 1991.

9. Hammond, K., and Hurwitz, N., Extracting diagnostic features from explanations, in *Proceedings of the Case-based Reasoning Workshop, 169–178*, Morgan Kaufmann, San Mateo, CA, May 1988.

10. Goel, A. K., Grounding case modification in deep models, in *Proc. AAAI Spring Symposium on Case Based Reasoning*, 41–44, 1990.

11. Braverman, M. S., and Wilensky, R., Towards an unification of case-based reasoning and explanation-based learning, in *Proc. AAAI Spring Symposium on Case-Based Reasoning*, 80–84, 1990.

12. Dutta, S., and Bonissone, P. P., Integrating case-based and rule-based reasoning: The possibilistic connection, in *Uncertainty in Artificial Intelligence*, Vol. 6, (P. P. Bonissone, M. Henrion, L. N. Kanal, and J. Lemmer, Eds.), North Holland, Amsterdam, 281–298, 1991.

13. Branting, L. K., Exploiting the complimentarity of rules and precedents with reciprocity and fairness, in *Proceedings of Case-Based Reasoning Workshop*, 39–50, Morgan Kaufmann, San Mateo, CA, May 1991.

14. Golding, A. R., and Rosenbloom, P. S., Improving rule-based systems through case-based reasoning, in *Proceedings of the National Conference on Artificial Intelligence*, 22–27, AAAI Press, Menlo Park, August 1991.

15. Rissland, E. L., Basu, C., Daniels, J. J., McCarthy, J., Rubinstein, Z. B., and Skalak, D. B., Interpreting statuatory predicates, A blackboard architecture for CBR: An initial report, in *Proceedings of Case-Based Reasoning Workshop*, 77–92, Morgan Kaufmann, San Mateo, CA, May 1991.

16. Oskamp, A., Walker, R. F., Schrickx, J. A., and Van Emden Berg, P. H., Prolexs, divide and rule: A legal application, in *Proceedings of the 2nd International Conference on Artificial Intelligence and Law*, 54–62, 1989.

17. Bonissone, P. P., and Dutta, S., Mars: A mergers & acquisitions reasoning system, in *Computer Science in Economics and Management*, Vol. 3, Kluwer Academic Publishers, Holland, 239–268, 1990.

18. Ruspini, E., On the semantics of fuzzy logic, Tech. Note #475, Artificial Intelligence Center, SRI International, Menlo Park, California, 1989.

19. Ruspini, E., Possibility as similarity: The semantics of fuzzy logic, in *Proceedings of the Sixth Conference on Uncertainty in Artificial Intelligence*, 281–289, Cambridge, MA, 1990.

20. Ruspini, E., The logical foundations of evidential reasoning, Tech. Note #408, Artificial Intelligence Center, SRI International, Menlo Park, California, 1987.

21. Ruspini, E., The semantics of vague knowledge, *Revue de Systemique*, X, 1990.

22. Bonissone, P. P., Plausible reasoning: Coping with uncertainty in expert systems, in *Encyclopedia of Artificial Intelligence*, (S. Shapiro, Ed.), John Wiley and Sons Co., New York, 854–863, 1987.

23. Pearl, J., Evidential reasoning under uncertainty, in *Exploring Artificial Intelligence*, (Howard E. Shrobe, Ed.), 381–418, Morgan Kaufmann, San Mateo, CA, 1988.

24. Dubois, D., Lang, J., and Prade, H., Fuzzy sets in approximate reasoning, Part 2: Logical approaches, *Fuzzy Sets Systems*, 40(1), 203–244, 1991.

25. Zadeh, L. A., Fuzzy sets, *Information Control*, 8, 338–353, 1965.

26. Zadeh, L. A., Fuzzy sets as a basis for a theory of possibility, *Fuzzy Sets Systems*, 1, 3–28, 1978.

27. Zadeh, L. A., A theory of approximate reasoning, in *Machine Intelligence*, (P. Hayes, D. Michie, and L. I. Mikulich, Eds.), Halstead Press, New York, 149–194, 1979.

28. Sugeno, M. (Ed.), *Industrial Applications of Fuzzy Control*, North Holland, Amsterdam, 1985.

29. Zadeh, L. A., Probability measures of fuzzy events, *J. Math. Analysis Appl.* 10, 421–427, 1968.

30. Smets, P., Belief functions, in *Non-Standard Logics for Automated Reasoning*, (P. Smets, A. Mamdani, D. Dubois, and H. Prade, Eds.), Academic Press, NY, 253–286, 1988.

31. Bonissone, P. P., Approximate reasoning systems: A personal perspective, in *Proceedings of the National Conference on Artificial Intelligence*, 923–929, AAAI Press, 1991.

32. Bonissone, P. P., Summarizing and propagating uncertain information with triangular norms, *Int. J. Approx. Reason.* 1(1), 71–101, January 1987.

33. Bonissone, P. P., and Decker, K. S., Selecting uncertainty calculi and granularity: An experiment in trading-off precision and complexity, in *Uncertainty in Artificial Intelligence*, (L. Kanal and J. Lemmer, Eds.), North-Holland, Amsterdam, 217–247, 1986.

34. Schweizer, B., and Sklar, A., Associative functions and abstract semi-groups, *Publicationes Mathematicae Debrecen*, 10, 61–81, 1963.

35. Schweizer, B., and Sklar, A., *Probabilistic Metric Spaces*, North Holland, New York, 1983.

36. Ferrara, R. C., *Mergers and Acquisitions in the 1980s: Attack and Survival*, (Series title: Corporate law and practice course handbook series; no. 558), Practising Law Institute, New York, New York 10019, 1987.

37. Michel, A., and Shaked, I., *Takeover Madness*, John Wiley, NY, 1986.

38. Rock, M. L., *The Mergers and Acquisitions Handbook*, McGraw-Hill, NY, 1987.

39. Dutta, S., and Bonissone, P. P., An approach to integrating diverse reasoning techniques, in *Proc. of General Conf. on 2nd Generation Expert Systems and their Applications–10th International Workshop (Avignon-90)*, 37–51, 1990.

40. Kuhn, R. L., (Ed.), *Mergers, Acquisitions, and Leveraged Buyouts*, Dow Jones-Irwin, Homewood, Illinois, 1990.

41. Bonissone, P. P., Gans, S., and Decker, K. S., RUM: A layered architecture for reasoning with uncertainty, in *Proceedings of the 10th International Joint Conference on Artificial Intelligence*, 891–898, August 1987.

42. Bonissone, P. P., Aragones, J., and Stillman, J., PRIMO: A tool for reasoning with incomplete and uncertain information, in *Proceedings of the 3rd International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, (IPMU 90), 325–327, Paris, France, July 2–6, 1990.

43. Duda, R. O., Hart, P. E., and Nilsson, N. J., Subjective Bayesian methods for rule-based inference systems, in *Proc. AFIPS 45*, 1075–1082, AFIPS Press, New York, 1976.

44. Heckerman, D., Probabilistic interpretations for MYCIN certainty factors, in *Uncertainty in Artificial Intelligence*, (L. N. Kanaal and J. F. Lemmer, Eds.), North-Holland, Amsterdam, 167–196, 1986.

45. Dempster, A. P., Upper and lower probabilities induced by a multivalued mapping, *Annals Math. Stat.*, 38, 325–339, 1967.

46. Shafer, G., *A Mathematical Theory of Evidence*, Princeton University Press, Princeton, NJ, 1976.

47. Jacobs, P., and Rau, L., Scisor: A system for extracting information from financial news, *CACM*, 33(11), 88–97, November 1990.

48. Blau, L., Bonissone, P. P., and Ayub, S., Planning with dynamic cases, in *Proceedings of Case-Based Reasoning Workshop*, 295–306, Morgan Kaufmann, San Mateo, CA, May 1991.

49. Bonissone, P. P., Blau, L., and Ayub, S., Leveraging the integration of approximate reasoning systems, in *Proceedings of 1990 AAAI Spring Symposium in Case-Based Reasoning*, 1–6, 1990.

50. Dutta, S., Approximate reasoning by analogy to answer null queries, *Int. J. Approx. Reason.* 5(4), 373–398, 1991.