N89-15574

Case-Based Reasoning: The Marriage of Knowledge Base and Data Base

Kirt Pulaski and Cyprian Casadaban

Martin Marietta Manned Space Systems
Post Office Box 29304 Mail Stop 3691
New Orleans, Louisiana 70189

Abstract

The coupling of data and knowledge has a synergistic effect when building an intelligent data base. The goal is to integrate the data and knowledge almost to the point of indistinguishability, therefore permitting them to be used interchangeably. Examples given in this paper suggest that Case-Based Reasoning is a more integrated way to link data and knowledge than pure rule-based reasoning.

1 Introduction

This paper describes some preliminary results of a NASA Mission Task being performed by the Automation & Intelligent Systems group at Martin Marietta Manned Space Systems in New Orleans, where the External Tank for the Space Shuttle is assembled. The goal of the project is to increase productivity at weld stations by decreasing downtime.

The plan to effect better productivity is to build an intelligent data base that gives advice about possible downtime causes and streamlines the follow-up paperwork. Efforts in the current fiscal year are producing a data base of reports of weld station downtimes as they occur. An ancillary knowledge base is growing as a result of the need to deepen the understanding of the weld station data.

A paradigm of reasoning needs to be selected that will best integrate the knowledge base and data base. Case-Based Reasoning (CBR) is being considered for several reasons:

CBR systems derive their power from knowledge base/data base interaction.

Both the knowledge and data are currently being collected case by case from the shop floor.

Advice-giving is memory-based and so is CBR.

Cases hold both data and knowledge in one structure, so the link between them is highly integrated.

CBR solves the problem of case disparity by being sensitive to and exploiting similarities.

The organization of this paper is to present the major components of the proposed Weld Intelligent Data Base (WIDB). First, a background of CBR is presented.

2 Case-Based Reasoning

The basic concept of CBR is simple: solve new problems by adapting solutions from old problems. The representation of a problem solving episode is called a **case**. Similarities between past and present cases establish a very high focus for problem solving. This focus is very difficult to achieve when only using rule-based reasoning (RBR).

CBR lends the power of examples to problem solving. Unlike the segmented explanations attributable to RBR, CBR explains its solutions with whole, relevant, concrete and familiar examples. CBR manifests learning as a by-product of adding cases, as they occur, to a case base. **Remindings** also provide a basis for knowledge acquisition (Riesbeck 88).

The generic system shown in Figure 1 depicts a cooperation between CBR and RBR (Pulaski 88). Each method has merits which the other lacks. RBR eases the implementation of a heuristic control strategy and is better for fast, non-complex, localized inferencing and for recording metaknowledge. CBR weaves the history of experience into problem solving.

3 Weld Data Base

When a downtime is reported a team of weld experts responds to the call. They work together to determine the cause of the problem, how to get the weld station operational as soon as possible and what to do to keep the problem from occurring again.

The result of a weld team call is a completed form which records, in several levels of detail, the path that the problem solving took from initial diagnosis through solution.

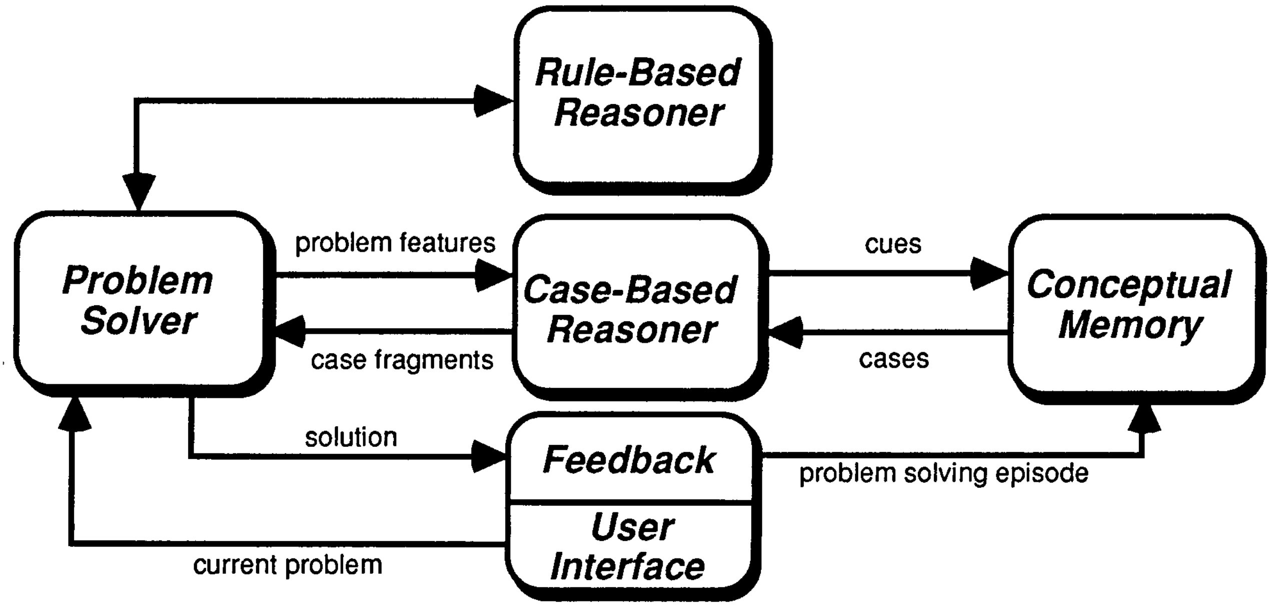

Figure 1: Generic Case-Based Reasoner

Each weld team report form is further broken down, analyzed and entered as a record into a data base. The year-to-date data base of weld station downtimes consists of about 250 records. Figure 2 shows an abbreviated version of a typical data base record that might result from a weld team call.

date	JAN-25-88
tool	T5018
effectivity	LWT-48
problem	TORCH-DOVE
cause	UNEVEN-TACK
downtime	H4
prepared-by	UNKNOWN
code	C88-21

Figure 2: Weld Team Report Data Base Record

4 Weld Knowledge Base

Figure 2 shows data from a weld team report. Certain types of knowledge need to be associated with the various fields, and the values in them, to enable an intelligent computational process to reason with that information.

One type of data base field knowledge is a systematic breakdown of allowable values for a given field. This knowledge is implemented as a **BNF grammar** which reduces the free English form to a parsed field value.

For example:

"No power to control system console."

is transformed into

CONTROL-SYS/CONSOLE/POWER/NOT/EXIST

according to the syntactic categories

System/Component/Subcomponent/Modal/Action

dictated by the problem description grammar.

The grammars do more than constrain field values. The grammar transformation rules can be used to support other functions such as form-filling and natural language (see Section 7). Also, the tokens of the grammars are often the same as the indices which cases in the case base are stored by, providing deep knowledge for storage and retrieval of cases.

Another type of knowledge relates field values to other field values. This knowledge is implemented as relational links in a field value hierarchy. Figure 3 shows part of the **Problem Tree**. The relational links are **ISA links**. For example, a PILOT-ARC-PROBLEM **ISA** TORCH-PROBLEM **ISA** PROBLEM.

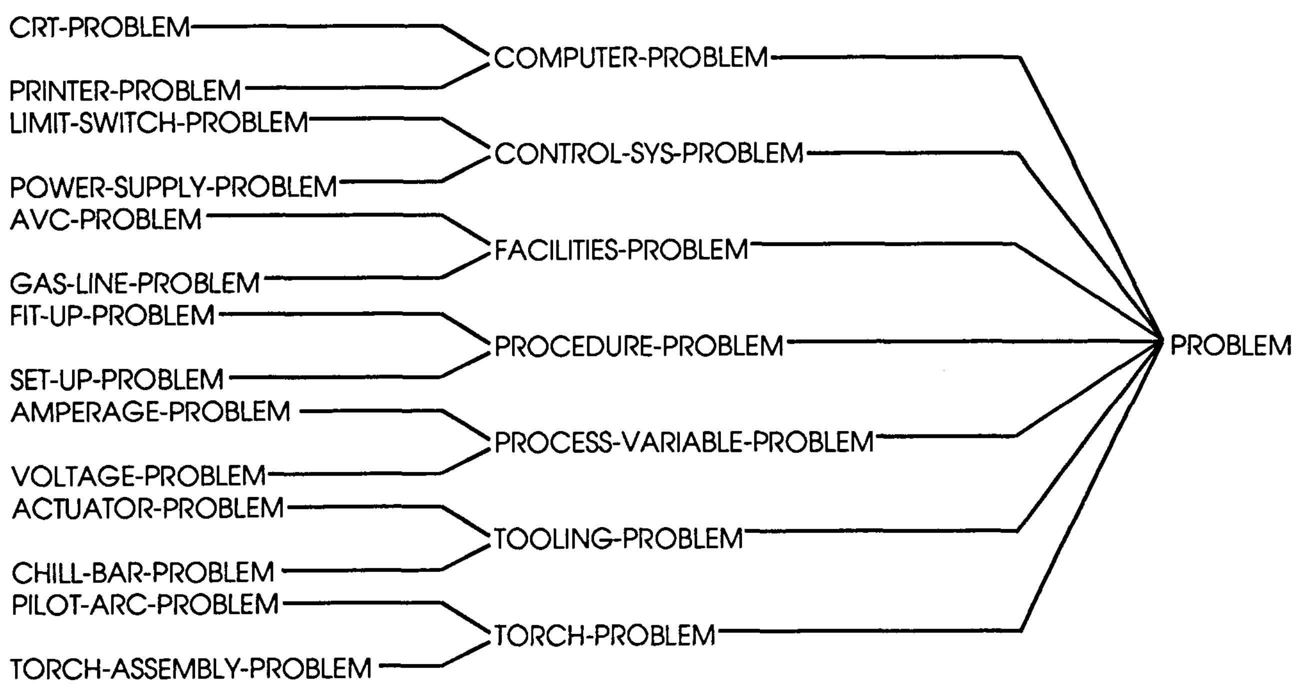

Figure 3: Problem ISA Tree

Relational knowledge helps to hypothesize uncertain or unknown knowledge in the data base. For example in Figure 2, the name of the person who prepared the weld team report is **UNKNOWN**. Using relational knowledge in the Problem Tree and the **Prepared-by Tree** helps to determine that the person who prepared the report was probably someone who usually handles **TORCH-PROBLEM**s.

Other relational knowledge can be used to link field values. For example, a **CAUSED-BY link** helps to consider causes for a problem. Any cause related to the current problem by a CAUSED-BY link is considered as a probable cause for that problem. Figure 4 shows that whenever there is a **TORCH-CUTTING** problem then three causes are considered.

GAS-LINE-CAUSE
TORCH-CUTTING-PROBLEM —— CAUSED-BY ← **ORIFICE-CAUSE**
TUNGSTON-CAUSE

Figure 4: The CAUSED-BY Relation

5 Weld Case Base

The **weld case base** is a union of the weld data base and the weld knowledge base. The case base looks very similar to the data base but with an implicit deeper understanding of the values found in the cases. One source of the deeper understanding is the weld knowledge base. Another source is the knowledge that accumulates over time as reminded cases are analyzed and reasoned about during advice-giving.

Figure 5 shows an abbreviated version of a weld case from the case base. The case representation shows some of what CBR adds to problem solving. When **CASE-145** occurred, the CBR reasoner was **reminded-of** three previous cases which helped to postulate a set of **related-problems** to check for. Causes for the problem and related problems were hypothesized and investigated. Causes which were not substantiated were stored as **failed-causes** along with **reasons** why. The actual **cause** was substantiated and its **justifications** were recorded. Later, CASE-145 served as a **reminding-for** two other cases.

6 Weld Intelligent Data Base

The advisory function in the WIDB is directly supported by CBR. The design of the WIDB includes other functions that CBR does not directly support. The following paragraphs mention these functions and what approaches will be used to implement them.

The paper work currently associated with the data collection task will be automated. Human interface and form-filling methods will be used to accept user input, verify it, categorize it and build a case representation for the CBR advice-giver.

name	CASE-145
date	JAN-25-88
tool	T5018
effectivity	LWT-48
problem	TORCH-DOVE
down-time	H4
prepared-by	TORCH-EXPERT
code	C88-145
reminded-of	CASE-007 CASE-063 CASE-129
related-problems	AVC-PROBLEM WIRE-FEED-PROBLEM
failed-causes	CONTROL-SYS-CAUSE WIRE-FEED-CAUSE
reasons	VOLTAGE-OK WELD-BEAD-OK
cause	UNEVEN-TACK
justifications	MISMATCH-PROBLEM WELD-SEAM-PROBLEM START-UP-PROBLEM
reminding-for	CASE-158 CASE-240

Figure 5: Example from the Weld Case Base

Report generation using the data base will include summaries, statistical analysis, trending and relational analysis.

Relational analysis may be aided with the use of a knowledge discovery tool, for example **IXL by IntelligenceWare, Inc**.

The use of **similarity networks** (Bailey 88) is being considered to augment case generalization to improve relational retrieval. Better relational retrieval will improve advice-giving.

A natural language interface will ease querying, browsing and receiving advice.

7 Discussion

In an advice-giving domain it is very helpful to reason from past examples. CBR, by definition, is then a strong candidate for the WIDB domain, plus it has even more to offer. The work described here shows that CBR is especially useful when a solution involves an intelligent data base; that is, a data base cooperating with a knowledge base.

The performance of an intelligent data base depends on the level of integration between the data base and the knowledge base. CBR offers the highest level of integration since both data and knowledge are stored in cases and are not distinguishable.

The case base can grow in different ways. As more data and knowledge are acquired, more cases are built and added to the case base. Each occurrence of advice-giving is also a case, so the system grows each time it is used. More cases means better remindings for future advice-giving. This learning mechanism does not require the CBR reasoner to change; its performance increases as a result of better remindings. This caliber of learning is very difficult to achieve with pure rule-based systems.

There will be times when a CBR reasoner cannot solve a problem or subproblem. As Figure 1 suggests, a rule-based reasoner is then used to generate the unknown solution. This also leads to learning since the solution is saved in the case base and the rule-based reasoner never has to solve that problem (or other problems like it) again.

8 Conclusion

The wide-spread acceptance of a knowledge-based technique into a mainstream computing environment depends on several important issues: improvement, embedability and integration. These issues are important because knowledge-based techniques rarely produce an entire solution to a problem. Rather, the best solutions piece together a mix of subsolutions which use both knowledge-based and conventional approaches.

The design of the WIDB addresses these issues. First of all, a knowledge-based technique must offer an **improvement**. The WIDB augments a conventional data base with knowledge. The improvement is a better and deeper working understanding of the information in the data base; a result of linking data base field values to eachother with knowledge.

The WIDB uses CBR for advice-giving type problem solving. This offers an improvement since conventional methods have difficulty implementing advice-giving which is sensitive to a history that is constantly being modified.

Another issue is **embedability**: knowledge-based techniques must embed into the same environment as conventional methods. The knowledge that the WIDB uses to augment its data base is fully embedable into a relational data base environment. The knowledge links that connect data base field values to eachother are implemented as relations (for example CAUSED-BY and ISA) in the relational data base itself.

The cases for CBR are also embedable into a relational data base environment. CBR provides a seamless link between a data base and a knowledge base; in fact, the cases in the WIDB hold both data and knowledge. Therefore, the cases can be implemented as data base records.

The last issue is **integration**. Different system components, whether knowledge-based or conventional, must fully integrate into one delivery environment. The WIDB integrates a CBR subsystem, a relational data base subsystem and a natural language front/back end. Each subsystem will be integrated with the others to operate in one delivery environment which is a 386-based microcomputer.

References

1. Bailey, D., Thompson, D., and Feinstein, J. "Similarity Networks." PC AI, July/August, 1988. pp. 29-32.

2. Pulaski, K. ELMO: An Episodic Long-term Memory Organizer for Case-Based Reasoning. AAAI Case-Based Reasoning Workshop, St. Paul, Minnesota, August 23, 1988.

3. Riesbeck, C. K. An Interface for Case-Based Knowledge Acquisition. In Proceedings of DARPA/ISTO Case-Based Reasoning Workshop, Clearwater Beach, Florida, May 11-13, 1988. pp. 312-326.