THE PACOR II EXPERT SYSTEM: A CASE-BASED REASONING APPROACH TO TROUBLESHOOTING

Charisse Sary
Computer Sciences Corporation
7700 Hubble Drive
Lanham-Seabrook, MD 20706

Abstract

The Packet Processor II (Pacor II) Data Capture Facility (DCF) acquires, captures, and performs level-zero processing of packet telemetry for spaceflight missions that adhere to communication services recommendations established by the Consultative Committee for Space Data Systems (CCSDS). A major goal of this project is to reduce life-cycle costs. One way to achieve this goal is to increase automation. Through automation, using expert systems and other technologies, staffing requirements will remain static, which will enable the same number of analysts to support more missions.

Analysts provide packet telemetry data evaluation and analysis services for all data received. Data that passes this evaluation is forwarded to the Data Distribution Facility (DDF) and released to scientists. Through troubleshooting, data that fails this evaluation is dumped and analyzed to determine if its quality can be improved before it is released. This paper describes a proof-of-concept prototype that troubleshoots data quality problems.

The Pacor II expert system prototype uses the case-based reasoning (CBR) approach to development, an alternative to a rule-based approach. Because Pacor II is not operational, the prototype has been developed using cases that describe existing troubleshooting experience from currently operating missions. Through CBR, this experience will be available to analysts when Pacor II becomes operational.

As Pacor II unique experience is gained, analysts will update the case base. In essence, analysts are *training* the system as they learn. Once the system has learned the cases most likely to recur, it can serve as an aide to inexperienced analysts, a refresher to experienced analysts for infrequently occurring problems, or a training tool for new analysts.

The Expert System Development Methodology (ESDM) is being used to guide development.

Pacor II Overview

The Pacor II DCF acquires, captures, and performs level-zero processing of packet telemetry for spaceflight missions that adhere to communications services recommendations established by CCSDS. Pacor II provides three forms of service for packet processing: real time, routine production, and quicklook. It strips packets from telemetry frames, reassembles packets, sorts packets by selected fields, merges packets from different sessions, and delivers scientific data sets and other related products to the user.

Analysts provide packet telemetry data evaluation and analysis services for all data received. Data passing this evaluation is forwarded to the DDF and released to scientists. Through troubleshooting, data failing this evaluation is dumped and analyzed to determine if its quality can be improved before it is released.

A major goal of the Pacor II project is to reduce life-cycle costs. One way to achieve this goal is to increase automation. Through automation, using expert systems and other technologies, staffing requirements will remain static, which will enable the same number of analysts to support more missions.

Problem Identification

Through discussions with Network and Mission Operations Support analysts, additional candidate areas for automation were identified. We focused on areas where the human reasoning processes of experts could be automated. Analysts provided a study that showed where they spent their time in the Hubble Space Telescope (HST) DCF for a 1-week period. Fifteen tasks were identified. The study described the percentage of staff-hours expended in each task for current operations and for projected future operations as workloads are expected to increase. The troubleshooting/dump analysis task had the highest potential benefit and was also suitable for implementation as an expert system.

Benefits

Through additional discussions with analysts, the troubleshooting problem was further evaluated for implementation as an expert system. Several potential benefits appeared to be possible.

Capture and store experience: Analysts felt that it would be useful to have a system that would enable them to more readily access prior troubleshooting problems and solutions. Currently, when problems recur, analysts must remember how they were fixed. If it is a problem that another analyst handled, analysts have to discuss it with each other or look up the problem and solution in a log book. Log books are available for analysts to record how they fix problems; however, specific requirements for the information stored there does not exist. The information may be sketchy, inconsistent, and difficult to find.

Analysts felt that a record of their prior troubleshooting knowledge, with an easy way to access the information, would help them in solving new or recurring problems. They also felt that troubleshooting experience from prior missions, including Pacor I, would be beneficial for Pacor II analysts at the start of the Pacor II mission, even though some problems may be new.

Expertise available during off hours: Shift analysts are the first analysts who fix problems that occur. If these analysts cannot fix a problem, troubleshooting analysts fix the problem. However, troubleshooting analysts only work during the day shift. An expert system could be an assistant to shift analysts on other shifts who do not have access to troubleshooting analysts and who are not as proficient in fixing problems.

Retain expertise with high turnover rate: Due to the nature of operations, analysts are required to work rotating shifts. Because this is demanding on the individuals involved, analyst turnover is high, which results in a high demand for training of new analysts. Analysts felt that it would be useful to have a system that would help in training and assisting inexperienced or new analysts perform their jobs. Also, because the Pacor II lifetime is expected to be long, expertise can be retained during personnel turnover through the use of expert systems.

Increased workload for same number of staff: Facility personnel currently handle complex decision-making processes. Through the use of expert systems, some of these processes can be automated, which frees the analyst to concentrate on exceptional situations and relieves the analyst from performing the more routine decision-making tasks. This automation would enable the same number of analysts to handle an increased workload.

Case-Based Reasoning Overview

CBR is a kind of expert system or another way besides rules to build an expert system. CBR uses past experience in solving new problems by storing previous experience or cases in a case base or database of cases. Cases are indexed so that they can be easily retrieved from the case base, and retrieved cases can be adapted to solve new problems.

Figure 1 illustrates the CBR process. Application domain knowledge is stored as a set of cases that describes past experience. Each case is composed of a set of features with values associated with these features. Typical information that might be included as features of a case are a description of a problem, a solution for the problem, how the solution was reached, and the expected result following implementation of the solution. Most often, the case base is developed incrementally over time as users find and solve new problems.

When a new problem is encountered, an analyst enters the characteristics or symptoms of the new problem as a new case. The CBR system searches the existing case base for cases that match and then displays a set of closely matching cases. Cases are ranked to indicate the degree of match between an old case previously stored in the case base and the new case.

If there are no exact matches, adaptation is often performed where a closely matching case is adapted to fit the new situation. There are two types of adaptation: manual and automatic. In manual adaptation, a user modifies a closely matching case manually. The modified case is then stored so that it can be reused when the problem occurs again. In automatic adaptation, the system automatically adapts an existing case. This adaptation is typically performed using a set of rules that describe how an existing case should be adapted.

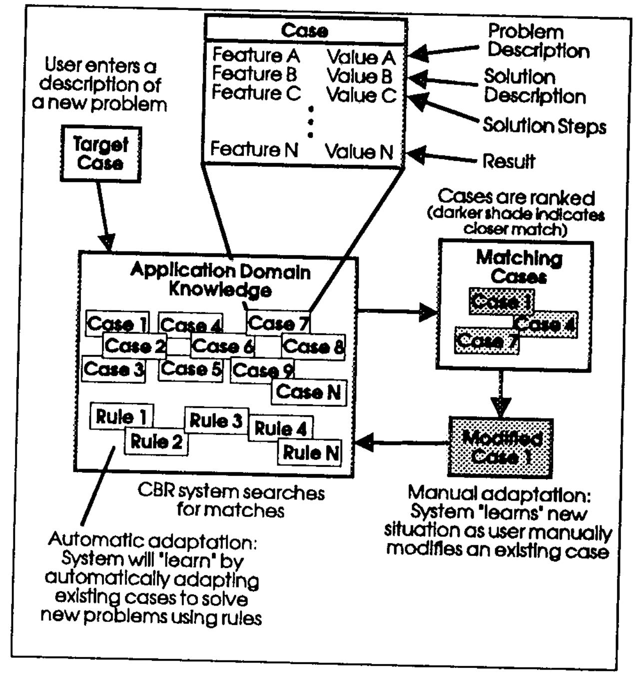

Figure 1. CBR Approach to Problem Solving

Advantages to CBR Approach

The CBR approach to problem solving has many advantages. Solutions to problems can be quickly derived because past experience is applied to the current problem. Previously obtained solutions can be reused rather than repeating the entire reasoning process each time the same problem recurs. Novices can use a CBR system to quickly obtain solutions to problems without a deep understanding of the process involved in deriving the solution. Also, with CBR, novices are prompted for the important features and do not have to remember what is important, which makes CBR systems useful training tools. Finally, past correct solutions and solution paths, as well as past mistakes that may have been forgotten, can be reapplied to new problems, eliminating "reinventing the wheel." The system becomes more robust as more cases are added or existing cases are modified.

Rule-based expert systems have been widely used to handle problems dealing with automating the human reasoning processes of experts. The CBR approach to problem solving has many advantages over the rule-based approach. It is often easier to add new cases to a case base as compared to adding new rules to a rule base. For example, it is not always clear what the effect of adding one rule to a rule base will have on other rules in the rule base. In CBR, each case is an independent entity and does not interact with other cases as a rule does when it fires other rules.

CBR solves problems more similarly to the way humans solve problems. Humans most often use what they already know in solving a new problem, reapplying a previous solution path and solution, rather than generating a new solution every time. They adapt what they already know to solve a current problem. Because cases are more understandable to the end user or expert, CBR systems are easier for a human to understand, build, use, and maintain, which also makes knowledge acquisition easier. However, as with any intelligent system, users must be cautioned not to blindly apply the recommended solution without thoroughly evaluating it to ensure that it is indeed the correct one.

Two types of problems are most suited to the CBR approach: (1) those where a significant number of past experiences or cases are available that are applicable to new problems and (2) problems where all solutions or expertise are not known in advance or where the domain is not well understood.

Rationale for Choosing CBR

Based on the characteristics of the troubleshooting problem, we felt that the CBR approach was a suitable approach for troubleshooting for several reasons. Pacor II conventional software is under development. Therefore, the necessary troubleshooting expertise for Pacor II does not currently exist. However, a troubleshooting assistant could be developed for Pacor II analysts from existing mission experience and, subsequently, for logging Pacor II troubleshooting sessions after Pacor II becomes operational. A Pacor II troubleshooting system could be developed incrementally as knowledge is gained. Also, analysts could take a major part in populating an initial case base during development, after case base design is stable, and they can perform their own maintenance during operations.

Methodology

ESDM describes a standard methodology to follow when developing an expert system. Because requirements are unknown at the beginning of an expert system project, by developing a series of progressively more complex prototypes, requirements will be identified and validated. ESDM is based on an iterative life-cycle model or spiral model. Each iteration adds knowledge about what the human expert does and what the requirements should be for the system. Each iteration also reduces the risks and uncertainties about the feasibility and practicality of using expert system technology for a given system.

ESDM is composed of five stages. The product of each stage is an executable prototype. We are using ESDM for this project and have developed the first-stage prototype or a Feasibility Stage prototype.

The prototype produced during the Feasibility Stage automates one or a few key functions of the human expert and concentrates on feasibility issues.

Prototype Implementation

We have developed a proof-of-concept prototype that assists analysts in troubleshooting data quality problems. If the quality of the data received in the DCF is below a certain level, the analyst must determine the cause of the problem and decide if the quality of the data can be improved before it is forwarded to the DDF and to scientists.

The initial prototype is composed of a set of 12 cases. We expect the final system to contain about 100 cases. The cases range in level of detail from very broad, network-type anomalies to very specific, spacecraft-related anomalies. Categories of cases were classified into four general types:

- Spacecraft problem or spacecraft to ground station link problem
- Ground station to NASA Communications (Nascom) (GSFC) link problem
- Nascom to GSFC Building 23 inter-building data distribution resource/interbuilding data transmission system (IBDDR/IBDTS) link problem
- BDDR/IBDTS to Pacor II link/Pacor II internal problem

The initial case base contains cases from the first three categories. Six of the cases are from Pacor I and six are from the HST DCF.

Each case is composed of a title to identify a case, a set of symptoms or a description of the problem, a description of the cause of the anomaly (solution description), and an explanation of what an analyst should do to handle the anomaly (action). Figure 2 provides a sample case.

Title: Nascom to Sensor Data Processing Facility (SDPF) Link Problem
Problem Description:
Frame-level errors—*Cyclical redundancy code (CRC)*
Block-level errors—*Polynomial errors*
System results match—*Generic Block Recording System*
Packet errors—*Missing packets or gaps*
Percent recovery—*Greater than 100%*
Data Type—*Playback Recorder*
Data Inversion Performed—*No*
Gap characteristics—*No gap in block time*
100% recovery—*Yes*
Inversion flag changes and frame synch pattern is valid but inverted—*No*
Duration of gap—*Less than 4 minutes*
Number of missing packets—*Greater than 1*
Frame CRC corresponds to each packet gap location—*Yes*
Location of frame errors corresponds to location of block errors—*Yes*
Solution Description: Link problem between Nascom and SDPF
Action: Notify the Payload Operations Control Center and request a retransmission from the ground station. Request Nascom support for line checkout.

Figure 2. Sample Case

To match a new case with a case stored in the case base, a similarity assessment technique must be defined. In the prototype, the similarity between two cases is calculated by generating a score that indicates the normalized sum of the number of features that match between a new case and a case stored in the case base. Features that describe the symptoms leading to a problem are used in generating this score.

Figure 3 illustrates a sample prototype screen. At the top of the figure, an analyst has entered the characteristics of a current acquisition session. All of the closely matching cases retrieved from the case base are displayed at the bottom. Each line contains a score that indicates the degree of match between the current case and a stored case, the name of the matching case, and a brief description of the problem causing the anomaly. An analyst may retrieve a stored case from the case base and compare it to the case describing the current situation.

We currently use manual adaptation. If no exact matches are found, an analyst reviews the cases provided to see what other analysts have done in the past and decides if any of the proposed solutions are applicable to the current situation. If this is a new problem, an analyst may build a new case by entering the characteristics of the new problem, including the proposed solution. Later the solution may be verified or changed to a better solution, other incorrect solutions that were tried and discarded may be added, or alternate suitable solutions may be added.

Tool Chosen

The prototype was developed using the ESTEEM CBR tool, developed by Esteem Software Incorporated. ESTEEM is a

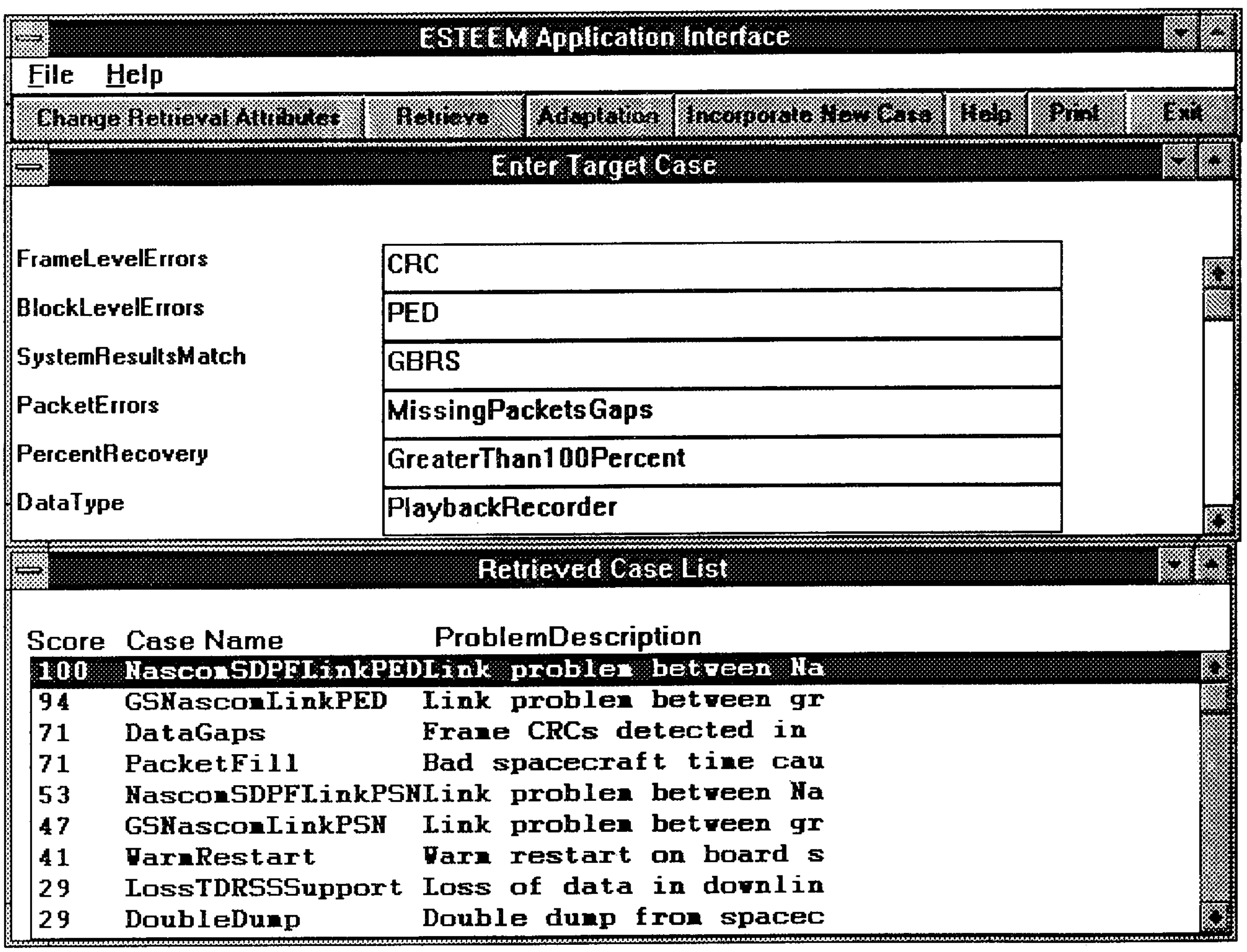

Figure 3. Sample Screen

standalone tool that runs on an 80486 IBM-compatible PC with 16 megabytes (optimal, 4-megabyte minimum) of memory, 5 megabytes of hard disk space, and a VGA monitor.

Future Issues

A major result of prototyping was to uncover issues that must be addressed in subsequent work. During maintenance in the operational environment, many analysts will have access to the case base. It needs to be determined if all analysts or if only the most experienced analysts will be permitted to add new cases to the case base. Also, it is very likely that analysts will have differences of opinion concerning the correct problem resolution. It needs to be determined whether all possible solutions or the most popular solutions will be added. Having alternatives could prove to be useful for situations where a close match is not found and an alternative solution is more suitable.

It is expected that in the operational environment, cases will evolve over time. A solution that an analyst initially thinks to be good could turn out to be in error, or an alternative solution may be better. The CBR system must be capable of evolving through this process.

For the prototype, we defined a set of features that describe the characteristics of the problem, the recommended solution, and the actions for handling the problem. For subsequent prototyping efforts, we need to determine if this set of features is suitable for all types of problems that analysts typically handle and for new, not-yet-encountered Pacor II problems. We need to determine if other information might be useful, such as other solutions tried that proved inadequate, additional background information or definitions for the inexperienced analyst, diagrams on how to fix a problem, and steps to follow to uncover the problem. A small analyst team has provided the expertise to build our initial prototype. The prototype must be evaluated by other analysts.

Because the Pacor II environment is UNIX based, we plan to port the prototype to the UNIX environment. The operational system will run as a tool for analysts who will extract feature values directly from the Pacor II database to minimize operator input. The final system will generate trouble reports automatically following an evaluation. Subsequent efforts will also include extending the case base and upgrading the computer-human interface.

Conclusion

This prototyping effort represents a novel approach to solving the troubleshooting problem using CBR. With advanced technologies such as expert systems, more automation can be introduced into operations, thus reducing life-cycle costs. Expert systems have been developed to handle troubleshooting using the rule-based approach. However, due to some of the unique characteristics of the Pacor II environment, the requirements of operations analysts, and the shortcomings of rule-based systems, an alternative approach was tried. This paper describes an initial proof of concept for the troubleshooting problem using CBR. A significant result of prototyping has been to confirm our hypothesis—we feel that this approach is a viable one for the troubleshooting problem.

Acknowledgments

The author would like to thank analysts Tony Walsh, Wende Peters, and Mark Hilliard for their contributions as experts during development.

References

Barletta, Ralph (August 1991). An Introduction to Case-Based Reasoning. *AI Expert*, Volume 6, Number 8.

Kolodner, Janet (1993). *Case-Based Reasoning*. San Mateo, CA: Morgan Kaufmann Publishers, Inc.