DAD: A Real-Time Expert System for Monitoring of Data Packet Networks

Sameh Rabie
___drew Rau-Chaplin
Taro Shibahara

Introduction

Modern telecommunication networks are increasing in complexity due to the diversity of equipment being connected and the variety of services being offered. At the same time, competitive pressures are placing more and more demands on operating companies to maximize the efficiency of their personnel and the performance of their equipment. Despite the availability of computerized support tools, effective network management requires a high degree of operators' alertness, coupled with the assimilation and mastery of vast amounts of operation and maintenance knowledge. Knowledge-based tools can help operators manage their networks more effectively and cope with the information overload inherently present in these demanding environments.

Bell-Northern Research (BNR) has undertaken several projects exploring feasible opportunities of expert system technology in networks maintenance applications [1–3]. Other organizations also have been active in the field, both in the telecommunications domain [4–6], as well as other industrial applications [7, 8]. However, the majority of early systems dealt with some isolated aspects of the operation and maintenance of the target environment. Furthermore, most of these systems ran in either batch or stand-alone interactive mode, and did not deal with the real-time and dynamic nature of the system under consideration. In contrast, the present work takes an integrated view to the network operator's functions including monitoring of real-time events, assessing the consequences of developing problems, as well as taking necessary repair escalation actions.

The goals of the described project are twofold: first, to explore the use of expert systems techniques for developing advanced operation support systems, which help in managing large telecommunication networks; second, to build a useful tool to assist the operators of the Canadian National Datapac™ network cope with the rapid network growth and evolution. The remainder of this paper reports on our progress to-date in developing and evaluating an advanced prototype system which addresses these concerns. The next section describes the prototype's environment and each of its functional components. The sections that follow highlight the knowledge base design techniques used, and describe our testing methodology and the results of a recent technical trial. Finally, we conclude with a summary of our key findings and discuss their significance to the development of future network operation support systems.

Datapac™ is a trademark of Bell Canada.

Functional Description

The Canadian Datapac Network is growing at a phenomenal rate. It currently consists of 74 DPN™ packet switches [9], distributed over 19 sites, and supports more than 35,000 connections spanning Canada from coast to coast. Telecom Canada network operators at the National Data Network Control Center (NDNC) monitor the network 24 hours a day to ensure swift problem identification and resolution. With the assistance of personnel at the switch sites, these operators are responsible for problem identification, problem clearing, and extensive recording and tracking activities. The Datapac ADvisor (DAD) is designed to help the network operators maintain and improve Datapac's high availability in the face of a growing and diversifying network. DAD is comprised of four loosely coupled components: a monitor, a problem-clearing advisor, a trouble-ticket creation system, and a collection of network databases (Figure 1).

Depending on the operator skill level and preferred mode of operation, DAD components may be accessed separately or in an integrated value-added fashion. For example, an operator might start working on a manually-created trouble ticket and later access the advisor for assistance, or

DPN™ is a trademark of Northern Telecom.

0890-8044/88/0009-0029 $01.00 © 1988 IEEE

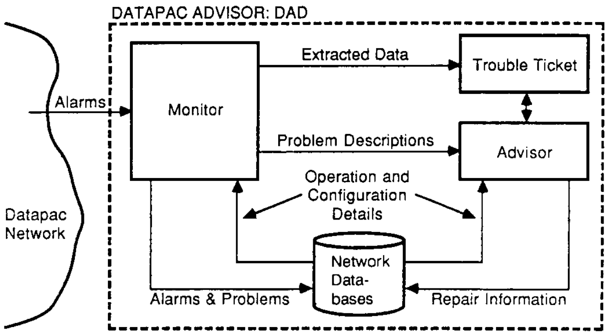

Fig. 1. Functional overview.

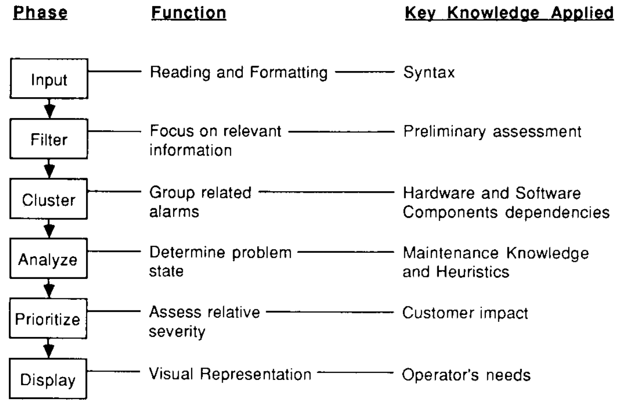

Fig. 2. Phases in monitoring.

alternatively, the operator might start out by selecting a problem flagged by the monitor, then invoke the advisor for assistance, and later conclude the session with a request for automatic creation of a trouble-ticket report. Network databases can be accessed either in a browsing context-free mode, or for context-sensitive retrieval, while using the other components of the system.

Besides the obvious advantages of system modularity and flexibility, the described architecture enables the creation of a "cooperative" man-machine interaction style, where both the operator and the system are in charge of what they are capable of doing best. DAD aids the network operator in identifying and solving network problems by providing advice, reminders, and background information as gleaned from a multitude of sources. However, the operator is left with ultimate control of the decision-making process; DAD is a decision support system, not a replacement for operators. It is our belief that the team, which is made up of a human network operator and an expert system, will be able to react and respond to situations faster and with more surety than either one could hope to do alone.

Monitor Subsystem

The aim of DAD's monitor subsystem is to respond to events and alarms as they occur on the network. It works directly in the real-time environment, analyzing the situation as each new alarm comes in. Each alarm is processed in six subsequent phases:

- Reading and formatting alarms
- Filtering out irrelevant information and redundant alarms
- Clustering together all alarms pertaining to a single problem
- Analyzing each problem to determine its nature, state, and progression
- Assigning priority to each problem
- Displaying the problem information to the operator

Figure 2 illustrates the six phases involved in monitoring and for each phase, lists the function and the key knowledge applied.

Graphical Problem Display

The monitor's display provides the operator with a clear, concise picture of any current network problems. In addition to identifying and analyzing network problems, DAD's monitor ranks them according to how critical and potentially costly they are. This allows network operators to devote all of their attention to a single problem, with the knowledge that the monitor will alert them should a more critical problem arise.

In addition to the alarm-driven monitoring which was previously described, DAD facilitates operator-initiated or active monitoring. In this case, the operator (perhaps based on DAD's advice) decides that a particular component should be periodically monitored to verify that it is performing correctly. Examples include post-recovery checking, and placing a "watch" on a troubled component or software queue.

In order to provide operators with a clear and concise picture of network problems, the monitor has been designed with an iconic interface. Each problem is represented by a display panel that identifies its location, type, importance, and exact situation. By selecting an individual icon of this multi-icon display, the operator can get further information relating to a particular aspect of the problem. Figure 3 shows the monitor displaying two concurrent problems.

Monitor Example

In Figure 4 the monitor display panel for a line processor failure is shown at four points during the line processor's recovery. Figure 4a indicates that Line Processor 6 on switch Belmont 4 has failed at 11:34. A warning icon informs the operator that the processor is currently in non-automatic recovery mode to allow corrective maintenance to occur should the processor fail. By selecting the Non-auto icon –N/A–, the operator will get a detailed list of the corrective actions that should be taken and the phone numbers of the node side to perform them.

In Figure 4b the processor has been returned to automatic recovery mode. The icons along the bottom of the display panel indicate that the processor and its first scanner have recovered and that the second scanner is expected

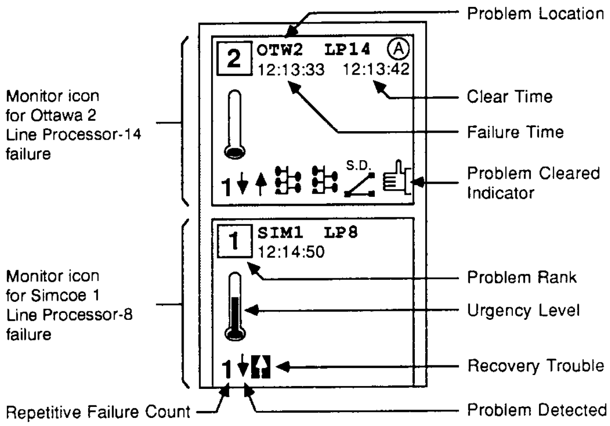

Fig. 3. Elements of monitor display.

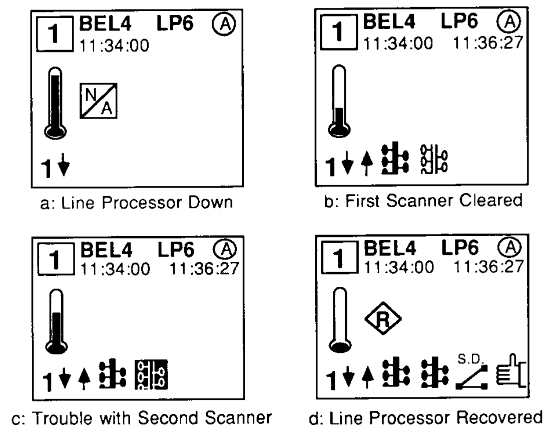

Fig. 4. Monitor example.

to clear shortly. The inverted display of the second scanner in Figure 4c indicates that a problem has arisen in its recovery procedure. By selecting this icon, the operator can see the alarms and any other pertinent information regarding this complication.

By the time we reach the state indicated in Figure 4d, the line processor has been successfully reloaded and recovered and has passed sanity checks as indicated by the "thumbs up" icon. The operator is also reminded of a current analyst's request for further information on failures involving scanner recovery problems. By selecting the request icon <R>, the operator can get detailed information regarding the data the analyst requires.

Advisor and Trouble Ticket Subsystems

The advisor's display consists of a trouble ticket region, a list of recommended actions, and an area for displaying the detailed procedures required to carry out an action. Figure 5 shows a snapshot of the advisor screen while solving a Don Mills 1 Line Processor problem. The operator determined the problem to be worked on by selecting the monitor's Don Mills 1 trouble displays. Notice that the ticket has been partially filled with data derived from the alarms; the alarm code has been interpreted as per the Northern Telecom practices; and a failure history for the processor has been compiled from the network databases.

The prioritized list of recommendations is ordered by three criteria:

- Most likely to solve the problem
- Fastest to execute
- Most likely to result in minimum service degradation

The operators are free to ignore any actions that they feel are inappropriate, or perform the actions in any order they please. If a more pressing problem develops on the network, the operator may select the new problem, returning to the current problem at a later time without losing its context.

One of the most challenging design aspects for expert systems working in real-time environments is to ensure that the monitor, advisor, and operator all work in synchronization and harmony. This is achieved in DAD by following a simple but effective protocol. The monitor will keep the network operator aware of any changes in the network state, but it will not update the advisor session until the operator has signaled his acceptance. For example, assuming that the operator invokes the advisor to help solve a problem, but the problem state changes halfway through the session because of autonomous network action, this protocol will ensure that the advisor session will not be swept out from under the operator's feet.

System Design and Implementation

The DAD system is implemented in KEE™ [10], a powerful expert system shell, running in a LISP environment on a high-resolution graphic workstation. This environment is designed to encourage rapid prototyping and exploration and to facilitate the development of advanced man-machine interfaces.

Since the system required the integration and representation of the many and varied types of knowledge used by operators in performing their job, no single knowledge representation scheme seemed sufficiently natural or expressive. Therefore, a multi-paradigm approach was selected, in which the paradigms were used where they were deemed most natural. For example, frames were used to represent system objects such as hardware configuration and problem categories. The frame paradigm offered several advantages in terms of modularity, information hiding, and property inheritance [11]. Rules, which are most effective in capturing heuristic knowledge, were used to represent the rules of thumb that our experts had developed through on-the-job experience. The procedural paradigm, in the form of LISP code and methods, proved useful in capturing the more procedural aspects of maintenance knowledge in addition to general system control. Demons were especially effective in managing the iconic displays, due to their highly dynamic nature.

KEE™ is a trademark of Intellicorp.

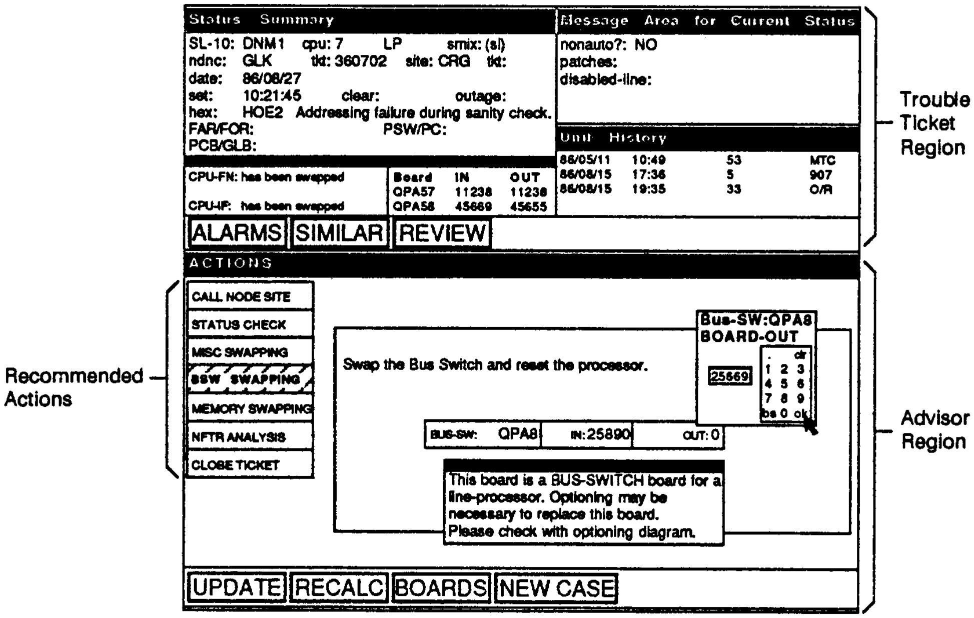

Fig. 5. Advisor and trouble ticket display.

Example of Objects Hierarchy

Figure 6 illustrates how DAD uses a hierarchy of objects that represent problem situations by inheritance. A problem classification hierarchy is used in the monitor to help recognize and analyze problems by interpreting the sequence and type of alarms received.

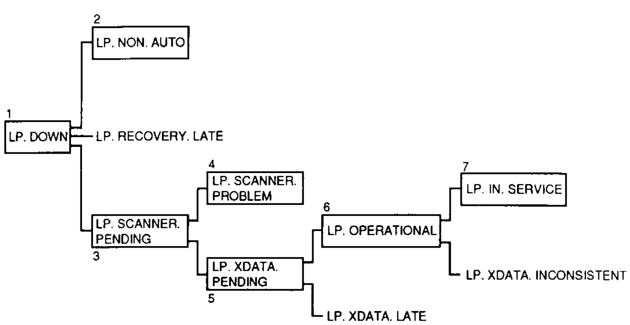

Fig. 6. Classification hierarchy (Line Processor Recovery).

A line processor failure would begin by being a member of the LP.DOWN classification. As a member of this class, it inherits situation-severity values, expectations regarding future events, and rules for identifying new situations that can evolve from the current situation. In the process of recovering, a problem would progress down the classification tree inheriting new values at each step. Eventually, it would become a member of the LP.IN.SERVICE class, indicating that the problem is cleared and the processor fully functional.

For example, the problem on Belmont 4 Line Processor 6 (described earlier) would begin as a member of LP.DOWN classification (1). It would then move to the LP.NON.AUTO classification (2), and inherit information regarding the corrective actions to be taken. Upon receiving a clear alarm, it would move to the LP.SCANNER.PENDING classification (3) and, in doing so, build up expectations for scanner clear alarms within an established time frame. In this case, a scanner problem alarm was received, causing the problem classification to move to LP.SCANNER.PROBLEM (4) and the priority of the problem to rise sharply as no further recovery is possible without operator intervention. From here the problem recovered normally, moving to LP.XDATA.PENDING (5) and on through LP.OPERATIONAL (6) to LP.IN.SERVICE (7) by successfully passing recovery checks.

Rules and Reasoning

Figure 7 is an example of a rule used by the monitor subsystem during the situation analysis phase. It updates the problem state after having received a scanner-clear message. As can be seen, rules provide the advantages of modularity and readability as each rule represents a self-contained chunk of knowledge. Both the monitor and advisor make use of rules for representing the expert's heuristic knowledge.

The expert system shell allows the specification of either forward chaining (data-directed), or backward chaining (goal-directed) reasoning styles. It was found, after some experimentation, that forward chaining was best suited for

```
If   (The COMPONENT of ?Alarm is SCANNER1)
And (The HEX of ?Alarm is FEFF)
And (The TYPE ?Alarm is CLEAR)
And (The NUMBER.OF.SCANNERS of ?Cluster is 2)
And (The LP.SCANNER0.CLEAR of ?Cluster is RECEIVED)
Then
(A RELATED.ALARM of the LP.SCANNER1.CLEAR of ?Cluster is ?Alarm)
And (?Cluster is in class LP.XDATA.PENDING)
And (The LP.XDATA.CLEAR of ?Cluster is PENDING)
And (No.Longer.Expect LP.SCANNER1.LATE on ?Cluster)
And (Expect LP.XDATA.LATE on ?Cluster within 40 seconds)
```

English Paraphrase

If you receive a clear alarm on the second scanner of a two scanner line processor then mark the scanner as cleared, move to the service data pending class, and expect service data to clear within 40 seconds.

Fig. 7. An example of a rule.

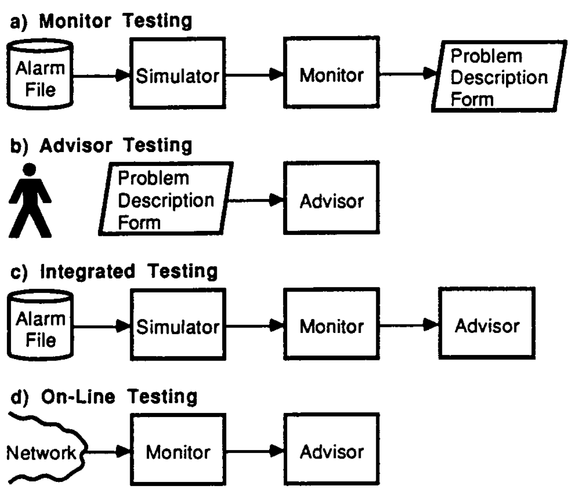

Fig. 8. Steps in testing plan.

the monitor's situation-analysis phase and the initial "make-hypotheses" stage of the advisor because it yielded all possible alarm interpretations and fault alternatives. Backward chaining, on the other hand, was found more appropriate for pursuing a top-ranking fault hypothesis and guiding the operator through associated repair procedures.

It is worth noting that the object-oriented paradigm was used advantageously in partitioning the rule base into rule contexts. This way, a relatively small number of rules had to be considered at any one time, which makes it possible to run the rule system in close to real-time.

Testing and Evaluating

Testing of a real-time and dynamic expert system like DAD poses an additional level of complexity beyond that for testing systems dealing with batch or static domains. The complexity comes from two sources: the difficulty of anticipating all the alarms and sequences of events that can occur during network failures; and the impracticality and error-prone nature of having the developers or testers manually type these alarms. The complexity of testing is increased by the fact that a yet untested system cannot be connected to a live network because of the stringent network reliability constraints.

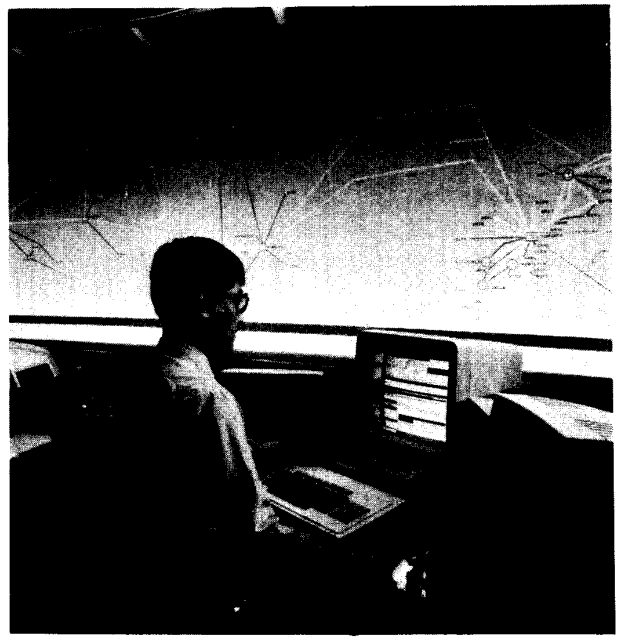

Fig. 9. The Datapac Advisor in action at the NDNC.

The architecture of DAD provides a narrow and well-defined interface between the monitor and advisor, which proved to be crucial for carrying out smooth and effective testing. This interface is implemented as an object (displayed as a text form) that contains a high-level abstraction of all the important problem features extracted by the monitor by sifting through the real-time alarm data. This form could either be read by testers to verify correct monitor operation, or alternatively, could be used for manual testing of the advisor operation.

As Figure 8 shows, the testing plan calls for the following phases:

- Testing of the monitor subsystem through reading files of alarm data in simulated real times
- Testing of the advisor through manual filling of the monitor output form, and conducting interactive repair sessions
- Integration testing of advisor and monitor through the use of alarm files
- Live testing of the system in the real network environment

The system underwent an evaluation trial at the NDNC during the fall of 1987. Figure 9 shows the prototype DAD in the NDNC environment. Users' reaction was very positive. Our initial analysis of the trial results indicated a 75%

success rate within the prototype's domain. These results confirm the soundness of the system concept and design approach. We are currently concentrating our efforts on increasing the depth and breadth of the prototype's knowledge, enhancing the performance of the real-time monitoring algorithms, and examining ways of embedding the prototype's software within future operation support systems.

Conclusions

Expert Systems offer significant potential for improving network management and operation efficiencies. In this paper, we presented a concrete example of a prototype expert system helping to manage the Canadian National Datapac Network. The project progress to date has been rapid and visible, resulting in increased user enthusiasm and cooperation.

Many of the concepts presented in this paper are generic in nature, and can set examples for building sophisticated maintenance systems in different communication networks environments. Examples of the applicable concepts include: the division of the system into loosely coupled cooperating components; the novel design and the use of iconic graphics in the monitor subsystem; the multi-paradigm knowledge representation approach used; and the cooperative man-machine interaction style between the system and the operator.

The potential impact of expert systems in the telecommunication domain is tremendous and by no means limited to maintenance. Other prime areas include operation, configuration, installation, provisioning, and planning. As our experience and ability to build these systems increases, we should expect to see more and more expert systems contributing to successful network operations and management.

Acknowledgments

We would like to extend our thanks to James Kennedy and Norm Smith, the project sponsors and coordinators from Bell Canada; and Dennis Parisienne, Chuck Dubson, and Norm Robertson, the Datapac domain experts from the National Data Network Control Center (NDNC) of Telecom Canada.

References

[1] T. Bult, D. Peacocke, S. Rabie, and V. Snarr, "An Interactive Expert System for Switch Maintenance," *Proc. of IEEE Intl. Switching Symposium*,ISS 1987.

[2] D. Peacocke and S. Rabie, "Knowledge Based Maintenance in Networks," *Proc. of IEEE Global Telecommunications Conference*, Globecom 1987.

[3] S. Rabie, T. Bult, V. Snarr Carter, and B. Cavan, "DRESS: Diagnostic Repair Expert System Shell," *IEEE Intl. Conference on Communication*, ICC 1988.

[4] G. Wrigley, "Expert Systems in Network Operation," *Proc. of IEEE ICC Workshop on the Integration of Expert Systems in Network Operations*, 1987.

[5] S. Goyal, D. S. Prerau, A. V. Lemmon, A. S. Gunderson, and R. E. Reinke, "COMPASS: An Expert System for Telephone Switch Maintenance," *Expert Systems*, vol. 2, no. 3, July 1985.

[6] G. T. Vesonder, S. Stolfo, J. E. Zielinski, F. D. Miller, and D. M. Copp, "ACE: An Expert System for Telephone Cable Maintenance," *Proc. Intl. Joint Conference on AI*, IJCAI 1983.

[7] W. Faught, "Applications of AI in Engineering," *IEEE Computer*, vol. 19, no. 7, July 1986.

[8] L. Pau, "Survey of Expert Systems for Fault Detection, Test Generation, and Maintenance" *Expert Systems*, vol. 3, no. 2, 1986.

[9] D. Jeanes and A. Khawar, "DPN: Leading the World in Packet Switching," *Telesis (BNR)*, vol. 14, no. 2, 1987.

[10] J. Kunz, T. Kehler, and M. Williams, "Applications Development Using a Hybrid AI Development System," *AI Magazine*, vol. 5, no. 3, 1984.

[11] R. Fikes and T. Kehler, "The Role of Frame-Based Representation in Reasoning," *Communications of the ACM*, vol. 28, no. 9, Sept. 1985.

Sameh Rabie received his Honors B.Sc. (1968) from Cairo University in Egypt, and his M. Eng. and Ph.D. (1973 and 1976 respectively) from McGill University in Montreal, Canada, all in electrical engineering.

Upon graduating, he first worked for the Systems Group of Canadian Pacific selecting and developing computer business applications, and later worked for IT&T as Principal Designer on a private branch exchange (PBX) project. He joined Bell-Northern Research in 1982 to work on the development of specification and emulation tools for new telecommunication equipment and office products. He was promoted in 1984, to manager of BNR's Artificial Intelligence (AI) department. Since then he has been leading the department's activities in the development of expert system applications for the maintenance and provisioning of circuit and packet switching networks.

Dr. Rabie has over fifteen published papers on topics varying from optoelectronic devices to user interfaces and expert systems. He is a member of the Association of Professional Engineers of Ontario, and the American Association of Artificial Intelligence.

Andrew Rau-Chaplin is a Research Scientist in the Data Packet Network AI Applications Group of Bell-Northern Research. His responsibilities include the application of expert systems technology to network operations and management systems, with particular emphasis on real-time environments.

He joined BNR shortly after graduating from York University with a degree in computer science. He has previously been involved with projects in advanced man-machine interface design, and programming environments for artificial intelligence research.

Mr. Rau-Chaplin is a member of the Canadian Society for Fifth Generation Research (CSFGR) and the Canadian Society for Computational Studies of Intelligence (CSCSI).

Tetsutaro Shibahara received an M.Sc. in electrical engineering from the University of Illinois in 1971 and a Ph.D. in computer science from the University of Toronto in 1985. His doctoral dissertation dealt with the use of causal knowledge for the recognition of time-varying signals such as ECG.

He is a Research Scientist in the Knowledge Technology Group of Bell-Northern Research. Prior to joining BNR in 1986, he was a research scientist of the Artificial Intelligence laboratory at the University of Toronto. He was also an Assistant Professor of the Information Science Laboratory for Biomedicine at Kyushu University Hospital, Japan.

His current interest is in designing and implementing of maintainable knowledge-based systems, in particular diagnostic systems for telecommunication networks, and signal/pattern recognition systems. Dr. Shibahara can be reached care of the Knowledge Technology Department, Bell-Northern Research, P.O. Box 3511, Station C, Ottawa, Ontario, Canada.