Image Cover Sheet

CLASSIFICATION	SYSTEM NUMBER 149700
UNCLASSIFIED	

TITLE

AN ANALYSIS TOOL FOR THE BACKBONE OF THE AUTOMATED DEFENCE DATA NETWORK

AN: 95-00539

System Number:

Patron Number:

Requester:

Notes:

DSIS Use only:

Deliver to: NL

An Analysis Tool for the Backbone of the Automated Defence Data Network

Karen Bryden[1] & Dwaine Wattie[2]

Abstract

An expert system is described which monitors a DND communications network, displays the status of the network and provides the best available advice on recovery and repair. The expert system represents a practical attempt to integrate an object-oriented relational database and an object-oriented graphical programming environment into a hybrid knowledge-based system. In this paper, the requirement and design of the expert system are discussed. As well, the integration of the database and knowledge base is discussed in terms of its motive, the potential benefits of such integration, the method in which it was accomplished, and finally the limitations of the method in achieving the desired benefits.

Introduction

The Automated Defence Data Network (ADDN) is a far flung message switching system which links elements of the Canadian Forces throughout Canada. Maintaining and operating this network poses a formidable logistics task for a number of reasons. Its sheer size and complexity requires a large well trained work-force. Built in the early part of the 1980's using then mature technology, it is not capable of being retrofitted to embed automated fault location and system restoral. Considerable expertise is required to operate the network and interpret system problems.

The Network Operations Advisor (NOA) is the second of two knowledge-based systems designed and deployed in the ADDN. Its job is to monitor the network, identify and log events affecting service and provide troubleshooting information to the operator. The scope of NOA encompasses the network at large and provides real-time data on the status of every major component. A future enhancement will allow commands to be sent directly to system computers to configure the network or perform diagnostics under operator control.

A high performance Graphical User Interface (GUI) was identified as a key requirement of NOA. In order to reduce the effort required for graphics development, a commercial object-oriented graphical programming environment called SuperCard® was adopted for the task. SuperCard® objects contain scripts which can be used effectively as rules, however the environment does not support frame-like knowledge representation. Representation of knowledge as frames was accomplished by adding an object-oriented relational database and merging the two into a hybrid knowledge- based system. The resulting system can be queried on-line by an external agent and its elements can be extracted for reuse in other systems.

The use of object oriented programming environments such as SuperCard® for construction of expert systems is well known [1]. The potential of modern relational database tools for representing domain knowledge has been investigated [2,3]. A rule base, stored as elements of a relational database, makes available the power of the relational data model for rule maintenance and browsing [4]. In addition, the coupling of database technology and knowledge-base technology is one avenue of current research on knowledge reuse and knowledge sharing [5].

In the following sections, the DND requirement is described, and the features of this requirement which had a major impact on the design of NOA are discussed. Following this, the operation of NOA is described briefly with an example. The integration of database and knowledge base within NOA is then discussed in terms of its motive, the potential benefits of such integration, the method in which it was accomplished, and finally the limitations of the method in achieving the desired benefits.

Problem Description

A system was required to oversee the operation of a wide variety of interactive subsystems which comprise the ADDN (Fig. 1). NOA would identify and provide advice on equipment and telecommunications problems, recognize network

1. Communications Research Centre (CRC), 3701 Carling Ave, Ottawa, Ontario K2H 8S2
2. Department of National Defence (DND)/Directorate of Information Systems Engineering and Management (DISEM 6-2-3), 257 Slater St, Ottawa, Ontario K1P 5H9

jeopardy situations and furnish recovery and repair procedures. It would additionally provide general background knowledge and information collection procedures. The end users of the system were to be network System Control Centre operators and node personnel. It was to be fielded on personal computers.

Because the ADDN carries classified traffic, a direct link between NOA and the ADDN was initially ruled out. However the benefit of having NOA's database automatically updated coupled with the operational advantage of using the GUI as a status-board was so compelling that a link was later added. However, the capability was left for data to be entered manually by the user.

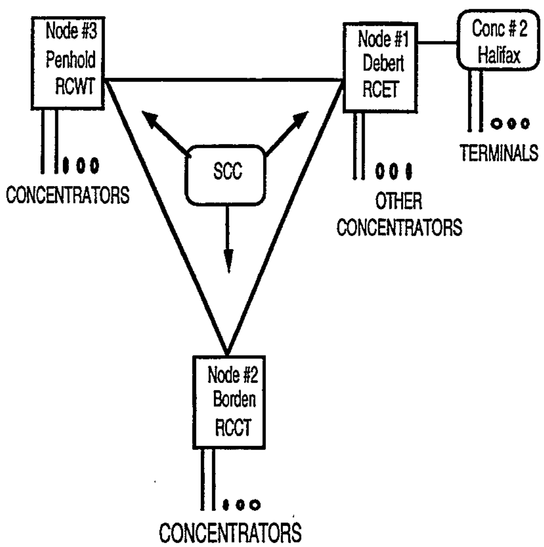

Figure 1 Automated Defence Data Network

Design Considerations

Initial experiments with a commercial expert system shell were not encouraging. The shell chosen was very slow on the Macintosh® host computer and a large development effort on the GUI appeared necessary. To avoid this, a combination of SuperCard® and HyperHIT-R[1]® was chosen, leading to the internal architecture of NOA shown in Fig. 2. SuperCard® is an object oriented programming tool with a graphical environment and superior programmability of colors of objects. HyperHIT-R® is a full-featured suite of relational database tools which can be called from SuperCard® scripts.

SuperCard®[2] scripts were found to be very capable as a medium for rules. Instantiation and unification (combinations of descriptive variables) are readily implemented, as are forward and backward chaining. Manipulation of the GUI is accomplished by SuperCard® scripts which respond to network events and actions of the user.

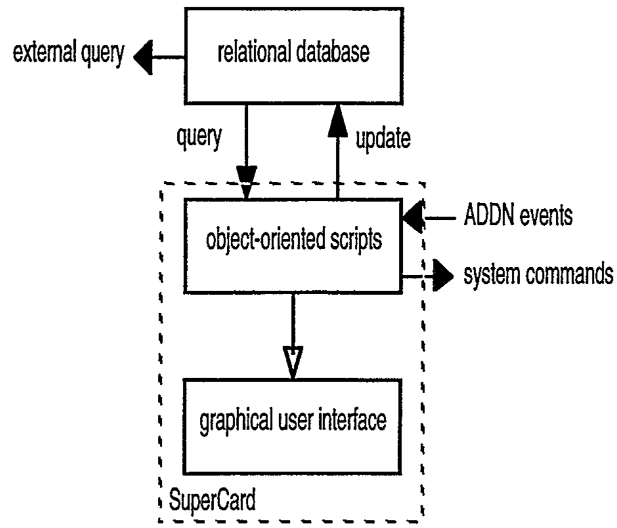

Figure 2, NOA Internal Organization

<u>Object-Oriented Database</u>

A relational database was used to record factual information in the database, including descriptions of problems and procedures, historical information on network outages and user notes. The database also represents aspects of the knowledge which cannot be readily represented in SuperCard®. Objects were created in the database representing the various subsystems and components of the ADDN, making it possible to use database queries in the reasoning process. Such an approach allows inheritance of properties, values and methods from classes to instances, and relations between objects to be explicitly represented.

In other efforts at integrating rule systems and database systems, the rules reside in the database. However this requires the domain knowledge to be encoded in the database query language, and severely limits the type of user interface which can be developed. In NOA, the database is copied into the

[1] Supercard is a trademark of Silicon Beach Software Incorporated, Aldus Corporation.

[2] Hyperhit is a trademark of Griffin Software.

user interface. This makes it possible to attach SuperCard® scripts to the objects and use the power of SuperCard® scripts and messages in order to reason on system events as well as operate the GUI. At system start-up, the GUI is rebuilt from the database, a necessary step to allow off-line maintenance of the database.

A separate SuperCard® database tool was created for browsing and editing NOA's underlying structure. It can be used for example to change the number or identity of terminals attached to the ADDN or examine the conclusions reached during testing of the expert system. With the multi-user version of the database, it is possible to open and query the database while NOA is operational.

Object-Oriented GUI

A GUI was required which would provide multiple views of the network at various levels of detail, and allow unconstrained browsing. The ADDN has a great deal of built-in redundancy. Communications paths can be reconfigured in a myriad of different ways to overcome an outage. A graphical representation of the network was therefore required which would accommodate all possible system topologies without being overly complicated.

This was solved by physically moving the objects around on the GUI like chess pieces. Objects which can be logically relocated such as concentrators and terminals are represented as list items in the GUI and are associated with corresponding records in the database. The use of list items and icons to identify and manipulate objects in the database is an essential feature of NOA.

Maintenance of Context

During ADDN operation, system tables are duplicated at each of the three nodes to identify the current topology of the network and thereby the routing required for individual messages. Configuration messages are sent among the nodes at regular intervals and also when necessitated by network events. These messages are intercepted and sent to the NOA host computer, where they are decomposed into one or more records, structured reports which drive the GUI.

During periods when configuration messages are arriving rapidly, it is possible for messages to be received out of order, conflict with each other, or alter the context of the user's current investigation. Because the GUI is shared by the network and the user, messages could potentially generate erroneous status, change the current context or focus on another aspect of the system. The result could be very confusing to the user. Therefore each message received must be checked for validity and conflict with current status. When a message updates the GUI, the users context must be saved and restored.

Real-Time Reception of Events from the ADDN

The link to the ADDN was written in C language and makes use of SuperCard's callable interface. It receives messages from the network on the Macintosh® serial port, searches for key words and assembles formatted records for each object affected. SuperCard® polls the record buffer, reads the records and processes them.

Expert System Operation

The top level of the GUI is shown in Fig. 3. From this view, the operator can examine the status of any node or concentrator in the ADDN. In this case, the operator will have noticed that the Penhold icon is yellow (indicating jeopardy) and clicking on the icon, will discover that DCP 1 (Data Communications Processor #1) has been deleted for 4 minutes. Using the pop-up menu at Penhold, the operator zooms in to obtain an internal view of the node. DCP 1 appears in red in this view and, clicking on the DCP 1 icon, the operator will observe that the result descriptor reported by the system is F000.

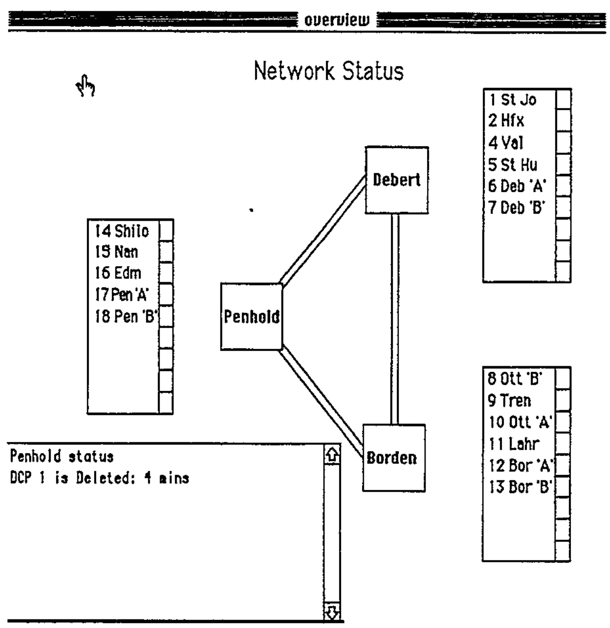

Figure 3, Overview of ADDN

Using the pop-up menu on DCP 1 to investigate DCP deletions, the diagnostic window shown in Fig. 4 will overlay the node view, giving an explanation of that result descriptor. By clicking on the topics, the operator can then browse at random through the best available advice on troubleshooting the problem. For example, clicking on "DCP Running?" and volunteering "No" to the question which appears will cause the answer to appear beside the question and provide a description of further tests. At any time, the operator can reconsider an answer, pursue another line of reasoning or defer the investigation while doing something else.

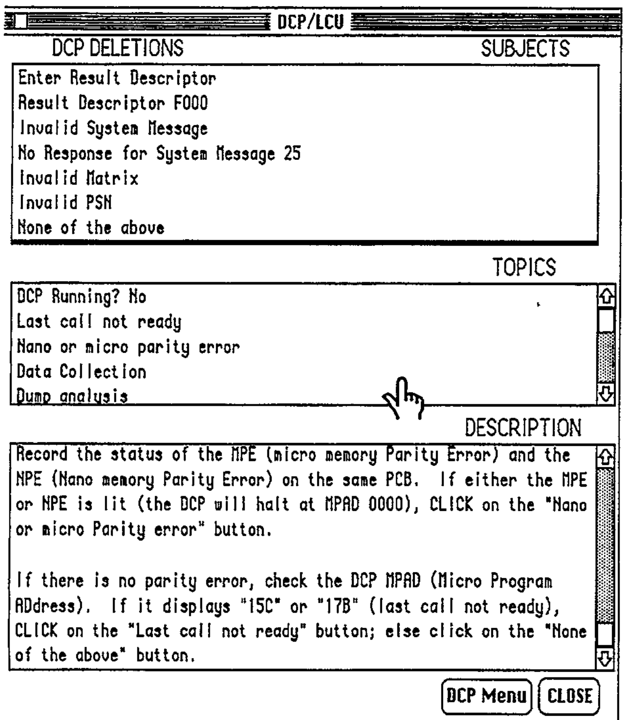

Figure 4, Analysis of DCP Deletion

Knowledge Management and Reuse

The domain of NOA encompasses that of the first knowledge-based system developed for the ADDN, the Concentrator Maintenance Advisor (CMA). The CMA was written in M1®. One task in developing NOA was to extract as much of the CMA knowledge base as possible, for reuse in the larger system. Only a modest part of the information could be extracted, working on the knowledge base with a text editor. Such experiences are all too common, and have led to efforts to create standards in the expert systems field.

There are several motives behind standardization efforts [9]. Among them, research will advance more quickly if expert systems can be built on top of existing knowledge bases. Knowledge could be shared among cooperating systems. Generic knowledge bases may form the basis of reusable knowledge libraries. Companies are seeking to manage their knowledge assets not only due to the high cost of their development but also the commercial advantage they confer.

Several approaches to the problem of standards are being pursued [6]. Among those under study are standards for knowledge representation such as the Initiative for Managing Knowledge Assets (IMKA) and methods for translating between knowledge representations such as the Knowledge Interchange Format (KIF). Studies are also in progress on run-time interfaces between object oriented databases and other intelligent agents, and interfaces between relational databases and knowledge representation schemes [7].

An attempt was made with NOA to integrate a relational database, a knowledge representation scheme using rules and frames and an object-oriented graphical user environment into a single functional unit. The advantage to be gained from this is the ability to use both knowledge base and database tools for problem solving. However a suitable external interface to the knowledge base offers possibilities for knowledge sharing and knowledge reuse (in the absence of current standards) without having to impose a fixed knowledge representation scheme.

In order to reap these benefits, the integration must take place in such a way that (a) the knowledge base elements are represented in the database and (b), the paths are reciprocal (eg. changes made to the database will be known to the inference engine). Furthermore, a database server must be developed which is capable of interpreting the internal representation of the knowledge. For example in [4], a knowledge dictionary was created which allows the power of the relational database model to be applied in the analysis of relationships among rules. In [8], the need for further research on rule "compilers" is shown as well as the need for more research on tools for supporting knowledge base queries.

Modern database tools make it possible to embed domain knowledge in the database. This was not considered a viable alternative for NOA, particularly due to limitations in the user environments. However in NOA, a practical degree of integration was achieved by duplicating objects in the database within the SuperCard® environment. The disadvantage of this approach is that in addition to increasing processing overhead, the inference engine is not notified of changes made to the database from the external interface. This limits the potential for an on-line database interface to a read-only status (although the limitation is not serious in this application).

Results and Summary

An expert system has been written which reads ADDN system event messages, displays the status of the ADDN in a multi-level GUI and provides a wide variety of information to the user, including the best available advice on recovery and repair. The GUI is completely non-modal, allowing the user to browse at random.

In the design of NOA, a relational database, a knowledge representation scheme and a graphical user environment have been integrated into a single functional unit with some limitations. This scheme provides a powerful development environment, and through the means of database query, allows much of the internal structure of NOA to be made available to an external agent. This capability can be used to allow NOA to provide read-only information to another system on-line, and allows the structures of NOA to be extracted for use elsewhere.

A primitive knowledge base browser built as an aid for knowledge base development allows the knowledge engineer to trace the inferencing process through the database by manual navigation. The object hierarchy as well as the current state and history of events of the ADDN can be extracted. A database server is needed to identify the links between the rules (which are partly explicit and partly built into the SuperCard® message hierarchy), interpret the linkages to objects, and explain what has happened as the result of an event. It remains to be investigated how to accomplish this task.

References

1. "expert system optimizes engine efficiency", The Motor Ship, June 1990.
2. R.E. Leroux and J.D. Wilson, "Data-Driven Expert System for PC Board Diagnostics", Proc. 3rd Symposium/Workshop on Applications of Expert Systems in DND, May 1991.
3. K.A. Sosin, E Marois and P.R. Roberge, "A Knowledge Based System for Battery Selection in Avionics", Proc. 3rd Symposium/Workshop on Applications of Expert Systems in DND, May 1991.
4. B. Jansen and P. Compton, "The Knowledge Dictionary: A Relational Tool for the Maintenance of Expert Systems", Proc. Intl. Conf. on Fifth Generation Computer Systems", Nov. 1988.
5. R.M. MacGregor, "Knowledge-Representation Standards Effort", 7th IEEE Conference on Artificial Intelligence Applications, Feb. 1991.
6. R. Neches et al, "Enabling Technology for Knowledge Sharing", AI Magazine, Vol 12 No 3, Fall 1991.
7. D. McKay, T. Finin and A. O'Hare, "The Intelligent Database Interface", Proc. Eighth National Conf. on Artificial Intelligence, 1990.
8. M. Stonebraker, "The Integration of Rule Systems and Database Systems", IEEE Trans Knowledge and Data Engineering, Vol 4 No 5, Oct. 1992.
9. C. Hall, "Why standards? The CAIA '91 Workshop on Standards for Artificial Intelligence", IEEE Expert, Dec. 91.