KNOWLEDGE-BASED SYSTEMS IN NETWORK MANAGEMENT

Technical Report No 106

T Wooding and J Malcolm

January 1990

Knowledge-based systems in network management

T. Wooding and J. Malcolm
Division of Computer Science, Hatfield Polytechnic, UK

Abstract

Many applications of knowledge-based systems (KBS) in network management have been described, though few have seen practical use. Most work has been in the areas of network design and fault diagnosis, using consultative systems. To apply KBS to the important task of network control will require a faster response time from large knowledge bases and greater trust of such systems by their users.

Introduction

Network management is concerned with the effective planning and utilisation of network resources. A network may connect a great variety of different devices and may itself consist of a number of different types of component. The network manager will need to have some knowledge of all of these. Networks are continually increasing in size and complexity. For example, a network may now connect a number of other networks, with a complex interconnection pattern, instead of just a number of devices. This means that a network manager, as well as having to cope with very diverse knowledge, may also have to deal with a huge amount of information. The increasing speeds of networks, together with the other problems associated with them may cause network managers to have to make several simultaneous decisions, often based on incomplete knowledge and within a considerably limited time period. Networks are also becoming increasingly critical for the operation of many businesses. Network management therefore seems to be an area where the application of KBS is particularly appropriate.

A number of KBS already exist for small isolated areas of network management, but the majority of KBS for this application area are still only research ideas or prototypes. This review attempts to categorise current work in this area and to suggest likely trends for the future.

The common trend in the development of software tools has been to advance from simple monitors, through advice-giving aids, towards action-taking experts. This trend can be traced through the development of tools for network management, starting with simple network monitors, moving onto advice-giving KBS, and towards the development of real-time network management centres which can both learn from experience and actually take actions based on decisions made. A more important factor in the acceptance and success of KBS in network management, however, is the realisation that different areas of network management place different requirements on the associated KBS. One requirement which varies considerably between the different areas of network management is response time.

The main application areas for KBS in network management are :

- design, including protocol design,
- planning and prediction,
- interpretation and diagnosis, and
- control.

There are no time constraints on design (other than the waiting customer), but as we progress down the list, time becomes increasingly important.

1. Design

Design uses both planning and prediction to produce a configuration of network elements based on user requirements (planning) together with historical data and/or performance trends (prediction).

The capture of user requirements and specification allow the designer to determine the best design to meet the customer's needs. The designer may then determine the characteristics of the network and set the parameters of the attached devices in order that the network can actually operate to meet the customer's needs. The area of network design thus incorporates the capture of user requirements, the specification of the whole network and it may also impact configuration management, while a separate, but related area is protocol design.

The process of network design is iterative and evolutionary. There are two stages to the process. The first stage involves the definition of the users' requirements and the specification of the total network, while the second stage is concerned with the more detailed design of the sub-systems.

Much of the information required for the less detailed stage is heuristic in nature, rather than technical, but few tools are currently available to aid the designer at this stage, which makes the task difficult. The heuristic nature of the information, however, indicates that KBS could provide appropriate tools. Similarly, one of the more detailed tasks, configuration management, is complicated by the size of the proposed network, the variety of components and the complexity of the connections between them.

The knowledge required for such a system is varied in nature. Although at a high level most networks are very similar and could therefore be represented by static structures, the structures, or some of their properties often need to be copied. For this reason frames are often used to represent the networks, because these provide mechanisms for replicating and/or inheriting information. On the other hand, application-specific design information is volatile and the best way to represent this type of information is by using production rules.

Response time is not important in this area of network management and so any response time limitations of KBS are irrelevant. In addition, a consultative system is all that is required to aid the task considerably, which means that a move towards a knowledge-based tool is more readily accepted.

An example of one KBS developed to aid the design area of network management is found in Designet (Mantelman, 1986). This system was developed for designing X.25 packet-switched networks, by Bolt Beranek and Newman (BB&N).

Designet includes a KBS as part of its simulation model, which is used to give the designer an idea of the traffic on the network as it has been designed. Designet can make suggestions on how to improve the model, which the user may choose to act upon or he may choose to input his own improvements.

In the current system there are four phases :

- the first stage involves the user inputting their requirements to the system,
- Designet then uses these to cluster the terminals attaching them to packet assemblers/disassemblers (PADs) or terminal-concentrating equipment,
- the third step involves more clustering. This time the PADs and hosts are linked to packet switches,
- then in the last step the packet switches are attached to a backbone network.

Designet displays its information in different formats - tabular, geographic or logical - to provide the user with different views of the network, depending upon which is most appropriate for the type of information concerned.

Heuristics are used to design the network, following the human expert's method of working and providing a minimum spanning tree just to ensure that all nodes are connected. The design can then be tested using a mathematical simulation of the network.

Other examples in this area are Expert Network Selector (ENS) (Ferguson et al., 1988), KDSS (Kinoshita et al., 1988), MAPCON (Muralidhar et al., 1988), NCONF (Thickett 1986) and NET/ADVISER (Mantelman 1986).

Protocol design, involves both analysis and synthesis to ensure the absence of logical errors, but both of these skills require experience. This area of network management has five main components :

- specification,
- validation,
- implementation,
- conformance testing of protocol implementation and
- conversion.

Here again the application of KBS is appropriate, because of their ability to handle diverse information and capture experience. An example system in this area is KSPS (Zhang et al., 1988), a KBS developed for protocol synthesis.

The knowledge-base contains five sets of rules :

- those for constructing explanations for why particular protocols have been developed,
- those for detecting buffer overflows,
- those for providing advice to the designer to help him decide if the information input corresponds to the components of the synthesis method,
- those for co-operating with the user interface and
- those for calling the external functions.

The type of inferencing engine proposed is a production system, because of the CAUSE....EFFECT nature of the reasoning used to design the protocols from the requirements.

The current development of KSPS can only handle specifications based on Global State Transition Graphs. This implementation also only covers the exchange of messages between two processes using two uni-directional, bounded, error free FIFO channels,

A more complex example in this area is Takahashi's system (Takahashi et al., 1988), a design for a complete support system for protocol development. This can use more formal specifications such as ESTELLE (ISO, 1986) or NESDEL (Shiratori et al., 1986).

2. Planning and Prediction

As part of a live network, planning and prediction can also be seen as a separate area of network management with tighter time constraints. In the event of failure of part of the network or one of its attached resources, a network manager may need to use a planning system to obtain advice on how to achieve a network with certain criteria. Similarly, he may need to use a prediction system to suggest possible consequences of actions he may be about to take to rectify a fault.

The lack of examples in this area supports the notion that the desired response time of a KBS affects the acceptance and development of such systems for network management. More than a consultative system is required in the dynamic environment, which is not as easily accepted or trusted, and is dependent upon the response limitations of KBS.

3. Interpretation and Diagnosis

Interpretation and diagnosis involve inference from sensor data and observables, by comparison with historical trends and/or predictions. This is useful in assessing network performance, fault isolation and recovery.

Networks are susceptible to a variety of faults. The type of fault which may be experienced might be the inability to make a connection or failure to transmit data. The faults when reported, either manually or automatically, should be interpreted by experts to detect and diagnose problems. In reality, however, troubleshooting is not this simple, although the actual task in identifying the faults is fairly repetitive. Hence the interest in KBS in this area.

Some of the problems are :

i. A single fault on the network usually creates more than one exception event. Detection and diagnosis of the fault must therefore involve a reasoning process which integrates the various events, correlating them on the basis of time and space and ignoring any redundant events. Both the time distribution and rate of events are important, because not only may related events be distributed over several days, but also a change in the network status, which happens only once an hour may not be important, but if it starts to happen every few minutes then it becomes more significant. It is also significant that certain events may only occur as a direct consequence of others, and may even occur after the problem has been fixed. Such events are redundant and should just be ignored.

ii. When a problem occurs on the network all the information required to solve it is very rarely available right from the start. Instead, the information is pieced together as the events arrive. This also means that hypotheses which were made early on may have to be changed in view of the new information.

iii. At any one time there are likely to be several unconnected problems waiting to be solved. Events relating to each of them will arrive interspersed with each other. Some problems will be more important than others and should be dealt with first, but while awaiting the arrival of the next event related to the high priority problem another problem can be considered. This requires some means of handling a number of problems simultaneously.

iv. The amount of information, including connection, configuration and protocol criteria, required to diagnose a fault is huge, which means that very few people know enough to carry out a diagnosis unaided. In addition, although each of the individual components might have its own set of diagnostic aids, there are very few tools aimed at diagnosing faults in the system as a whole. This means that an inexperienced engineer, trained only on the component-specific tools, who detects a system fault frequently has to refer the fault to an expert.

v. The continual change and evolution of networks and their components means that domain experience about obsolete components can be scarce.

vi. The number of events occurring in a network has been found to increase by N(N-1) where N is the number of connection points (Mathonet et al., 1987). This is significant because it directly affects the efficiency requirement of the reasoning mechanism which is important in developing a system which is designed to work in real-time.

A parallel can be seen between this problem domain and that of MYCIN (Shortliffe, 1976), in that both involve diagnosis of a problem requiring a large amount of diverse knowledge. MYCIN permitted the collection of expertise and a KBS solution for diagnosing network faults could provide the same type of help in the network management area.

The knowledge required for such a system falls into three categories :

- structural knowledge about the network and events,
- deductions, that is the data types created and manipulated during reasoning,
- knowledge about problem detection and diagnosis, specifying how to interpret events, recognise problems and isolate faulty components,

and means that a single paradigm cannot be used for adequate knowledge representation.

Response time needs to be kept to a minimum for KBS to be successfully applied to this area of network management. This is because interpretation and diagnosis is concerned with live, dynamic networks and so any delay in the problem-solving process directly affects the operation and full availability of the network.

There are several examples of systems which have been implemented and are actually being used. One of these is Automated Cable Expertise (ACE) (Vesonder et al., 1983; Wright et al., 1984; Mantelman, 1986; Zeldin et al., 1986), which provides trouble-shooting for telephone cable maintenance, by analysing maintenance reports produced by a data management and report-generation system, known as CRAS (Cable Repair Administration System).

Telephone cables are susceptible to a variety of faults including those of either an electrical or environmental nature and are mainly identified by customer-generated maintenance reports. Originally, the reports generated by CRAS were analysed by a number of experts to attempt to detect trends in the faults. The idea being to identify potential trouble spots, so that preventative action could be taken, before complete breakdown of the service occurred. ACE was developed because preventative action was not always being taken in time, despite the actual repetitiveness of identifying the faults. The problem was the huge volume of fault reports received and the lack of experts available to analyse them.

The characteristics of the domain suggested certain similarities with the domain of configuration, for which DEC's R1/XCON (McDermott, 1982) had been developed. The design of ACE was therefore considerably guided by the development of R1/XCON, in particular because the primary task of cable analysis, as with configuration, could be divided into a number of sub-tasks within a fixed sequence. This meant that similar to R1/XCON, there would be no back-tracking search of a large problem space, so the same problem-solving strategy based on production rules could be used.

The knowledge contained in ACE concerned the wirecentres, data and commands from CRAS, and analysis strategies used to interpret the information received. The division of the primary task into a number of sub-tasks in a fixed sequence meant that the knowledge-base could be loosely divided into sets of productions and related condition and action functions.

One set of productions examined the flow of maintenance reports each day. If a cable had no previous history then information was held indicating that it may require attention in the near future. In contrast, if a cable did have a previous history then additional information could be requested from CRAS. ACE then used this more detailed information not only to identify the trouble spot, but also to provide suggestions on how to carry out the repair.

Another set of productions contained the relevant knowledge to allow ACE to access CRAS, while yet another set of rules assembled the appropriate message about the day's events, recognised by the system and called on the the UNIX mail facilities to deliver the message to the appropriate users. This meant that the users were involved only if any trouble spots were found.

ACE has been in use since 1982, but although it performs the analysis quicker than a human analyst, no increase in the quality of service has been found. This is because ACE has to use data which has been stored rather than data obtained in real-time.

There are a number of other examples in this area - Brossier's system (Brossier et al., 1986), Central Office Maintenance Printout Analysis and Suggestion System (COMPASS) (Goyal, 1985; Goyal et al., 1985; Prerau et al., 1985a; Prerau et al., 1985b; Goyal et al., 1986), Cosic's system (Cosic et al., 1985), Communication Switch Maintenance Expert System (CSMES) (Harrington, 1986), Diagnostic Assistance Reference Tool (DART) (Bennett et al., 1981; Genesereth, 1984), Laffey's system (Laffey, 1986), MAD (Peacocke et al., 1988), NEMESYS (Guattery, 1985; Macleish et al., 1986; Macleish et al., 1987), NET/ADVISER (Mantelman, 1986), NETMAN (Zhan et al., 1988), Network Troubleshooting Consultant (NTC) (Mantelman, 1986), Switching Maintenance Analysis and Repair Tool (SMART) (Sutter, 1986; Mantelman, 1986), Testing Operations Provisioning Administration System - Expert System (TOPAS-ES) (Callahan, 1988), Troubleshooter (Mantelman, 1986) and Welin's system (Welin et al., 1986).

Two prototype systems have been found which make some attempt to respond in real-time. These are Datapak ADvisor (DAD) (Rabie et al., 1988) and DANTES (Mathonet et al., 1987).

In the case of DAD the identification stage occurs in real-time but the actual problem-solving stage remains under the control of the user. This means that the system continues to monitor the network while an existing problem is being solved. The user is informed if any other problems occur, but it is left to him to make the choice to do anything about this new information.

DANTES in addition to handling the identification stage in real-time, has also tackled the problem-solving stage. To approach real-time operation of this stage DANTES needed to minimise the amount of code executed and the number of disc accesses.

Most of the knowledge required for detection and diagnosis is heuristic in nature, which originally suggested the use of production rules for representing this knowledge. A production rule system, however would not have been very efficient when it came to the selection and application of the rules.

The alternative to rules is procedural attachment to structured objects, or demons, where the occurrence of an event triggers a demon to execute. This is an efficient method because selecting rules using state and trigger properties limits the number of rules tried at each step. Procedural attachement, however is not a very flexible method, so the compromise in DANTES has been to use a combination of the two methods. Consequently, a rule is defined as an object class and a rule application takes place for an instance of a class. A rule definition includes the class, rule name, trigger, state, environment and body. The trigger describes the event and the state defines the value of the object required for the rule to be applied.

The reasoning process involves events being received by DANTES and converted to messages for the network objects involved. At the message receptions the objects select the rules from the triggers. The rules are then tried in order of selection and it is from these rule applications that deductions can be created. The deductions in turn can trigger other rules, thus continuing the inference and updating the deductions network.

Time aspects in DANTES are handled in a number of ways. Knowledge about time correlation is represented as rules, whereas the revision of deductions with time is handled by adding a TIMEOUT viewpoint, containing all the properties for time management. A deduction is regarded as being in TIMEOUT if, and only if, it has not been IN USE for a predefined time. Similarly, a deduction is not IN USE during certain intervals if, and only if :

- it has not been used to derive any deduction(s),
- no deduction deduced from it is still IN USE.

When a deduction does TIMEOUT, the TIMEOUT process is reported to the associated object. The TIMEOUT viewpoint could be used to trigger rules just as events do. This allows for the removal of deductions after a given period, specific actions for deductions at regular times, and polling of certain conditions by the inference engine. This allows DANTES to deal with time aspects without increasing the time required for rule selection.

The number of disc accesses is limitied by limiting the virtual memory size and its usage. This has been achieved for DANTES by taking the following actions, but all aspects of memory management are machine dependent, so these actions might not have the same effect if the system were moved to another platform :

- deductions are allocated and deallocated by the inference engine in a special area where there is no garbage collection,
- the structural knowlege is static and can be loaded in a static area so there is no garbage collection,
- the parts (code or data) of the system which remain in physical memory have been selected to minimise disc access,
- the code used in the rule compiler has been reduced and optimised to aid efficiency and,
- the dynamic work area is quite small, with frequent on-line garbage collection.

One problem about this last point is that the number of deductions could become quite large. Although deductions which are no longer valid should be removed, it is important to record conclusions to problems because these are useful for direct diagnosis if a problem reoccurs repetitively on the same component. Selective garbage collection is therefore required and is achieved using either the TIMEOUT feature described above, or by removing all the deductions which are not used to deduce the given set.

DANTES has addressed a number of the problems associated with real-time KBS for network management, but the system is still dependent upon the efficiency of the implementation language, or tool.

4. Control

Control systems are responsible for ensuring that network performance is restored to an acceptable state as soon as possible. Again the use of KBS is indicated because of the volume of information to be searched and the fact that the knowledge required is often incomplete.

In control applications, a quick response time is crucial otherwise the network may go down completely. The optimisation of response time is dependent upon not only the development of fast KBS, but also the removal of any human intervention in the control loop, which is going to require considerable trust in the systems. This probably accounts for the small number and prototype status of systems in this area of network management.

One prototype system which has attempted to close the control loop and therefore optimise the response time is the Yorktown Expert System/Multiple Virtual Storage Manager (YES/MVS) (Ennis et al., 1986; Milliken et al., 1986). YES/MVS, however, only addresses one area of the management of a cluster of IBM mainframes, namely the problems associated with channel-to-channel links.

YES/MVS was designed for large installations, or clusters of IBM mainframe computers, each of which was controlled by the IBM Virtual Storage 2 Multiple Virtual Storage Operating System (frequently referred to simply as MVS). One purpose of such a large installation is to provide computing services to a group of submitted jobs under the control of the Job Entry Subsystem 3 (JES3) of MVS. Although there are normally a number of consoles for controlling the various components of the cluster, it is usually possible to perform both MVS and JES control operations from just one JES3 console. YES/MVS was developed to examine the problem of automatic control of a cluster of MVS/JES3 systems through one console, as well as providing advice on manual intervention if required.

Channel-to-channel linking is concerned with the networking of computers by means of I/O transmission links. If any of these links become inactive data traffic can be delayed and any queue space can become exhausted. It is therefore important that these links are monitored at regular intervals and if any degradation is observed corrective action is taken. This may involve freeing links from troublesome jobs, or re-routing data through other links and then attempting to restart the problem link if it has completely failed.

In the advisory mode YES/MVS presented any control actions, along with relevant explanations, to the operator to be checked before they were performed. If the operator agreed with the control actions YES/MVS would then continiue and submit the relevant command(s) to MVS. In contrast, the active mode included no such interaction with the operator, although an explanation of any actions taken could always be obtained.

The inference engine of YES/MVS was written using an extended version of OPS 5. OPS 5 is data-driven and therefore appropriate for a system designed to respond to information received from its target machine, but too slow to respond in real-time. To approach real-time control in YES/MVS, the right-hand side of the rules were compiled, LISP macros were used to optimise the matching process and the rules were distributed amongst several OPS 5 systems, on separate virtual machines supported by a host computer.

YES/MVS ran regularly for nine months at IBM Yorktown. During this time the KBS is reported to have sucessfuly detected and responded in the area implemented. The main criticism seemed to be that the system was too specific to one particular computer installation. Consequently, a decision was taken to build a second system which would be more like a shell, incorporating knowledge and services common to all control KBS and facilities to customise a KBS for a particular installation.

Other examples in this area are less advanced in their development, but include Automated Network Management (ANM) (Westcott et al., 1985; Feridun et al., 1988), Integrated Testing Expert System for Trunks (I-TEST) (Liu et al., 1988) and the Expert Telecommunications Resource Allocation Consultant (XTRAC) (Chuang, 1985).

Conclusion

The majority of KBS which currently exist for network management are consultative systems. Advancing technology, however, suggests that networks will inevitably continue to increase in speed, which means that the network management systems will be the determining factor in response time. A consultative system will therefore no longer be adequate, which would seem to indicate that fully automatic network management systems are required. It is not however, this simple.

Different areas of network management place different requirements on the associated KBS. In particular, areas such as design are not concerned with live networks and so there is no requirement for these systems to respond in real-time. Control systems, however, must respond as quickly as possible so that deterioration of the network is kept to a minimum. This means that for control there is a requirement on the system for it to respond immediately and therefore without human intervention. In turn, this dependency on response time determines the areas of network management where KBS are most readily accepted - design systems are the most readily accepted and control systems are the least.

References

Bennett, J. S. & Hollander, C. R. (1981) DART : An Expert System for Computer Fault Diagnosis. Proceedings of the IJCAI-81, Vancouver, Canada, 2, 843-845.

Brossier, J. Y., Faller, B. & Fron, A. (1986) A Communication Network : Trouble-shooting Expert Systems. International Conference on Communications ICC '86, Integrating the World Through Communications-Conference Record, Toronto, Canada, 2, 1238-1240.

Callahan, P., H. (1988) Expert Systems for AT&T Switched Network Maintenance. AT&T Technical Journal, 67(1), 93-103.

Chuang, D. (1985) Telecommunications Resource Allocation A Knowledge-Based System. IEEE Expert Systems in Government Symposium, McLean, Virginia, 666-675.

Cosic, C., Metzler, D. P. & Williams, J. (1985) Artificial Intelligence Techniques in the Detection and Diagnosis of Network Problems. ASIS '85, Proceedings of the 48th ASIS Annual Meeting, Las Vegas, US, 297-300.

Ennis, R. L., Griesmer, J. H., Hong, S. J., Karnaugh, M., Kastner, J. K., Klein, D. A., Milliken, K. R., Schor, M. I. & Van Woerkom, H. M. (1986) A Continuous Real-Time Expert System for Computer Operations. IBM J. Res. Develop., 30(1), 14-28.

Ferguson, I. A. & Zlatin, D. R. (1988) Knowledge Structures for Communications Networks Design and Sales. IEEE Network, 2(5), 52-58.

Feridun, M., Leib, M., Nodine, M. & Ong, J. (1988) ANM : Automated Network Management System. IEEE Network, 2(2), 13-19.

Genesereth, M. R. (1984) The Use of Design Descriptions in Automated Diagnosis. Artificial Intelligence, 24, 411-436.

Goyal, S. K. (1985) Expert Systems in Network Management. IEEE Expert Systems in Government Symposium, McLean, Virginia, 515-516.

Goyal, S. K., Perau, D. S., Lemmon, A. V., Gunderson, A. S. & Reinke, R. E. (1985) COMPASS : An Expert System For Telephone Switch Maintenance. IEEE Expert Systems in Government Symposium, McLean, Virginia, 112-122.

Goyal, S. K. & Worrest, R. W. (1986) Expert Systems in Network Maintenance and Management. International Conference on Communications ICC '86, Integrating the World Through Communications-Conference Record, Toronto, Canada, 2, 1225-1229.

Guattery, S. (1985) A Discussion of NEMESYS Implementation Issues. IEEE Expert Systems in Government Symposium, McLean, Virginia, 517-518.

Harrington, P. V. (1986) A Communication Switch Maintenance Expert System. GLOBECOM '86 : Communications Broadening Technology Horizons, IEEE Global Telecommunications Conference Record, Houston, US, 1, 0229-0233.

ISO (1986) ESTELLE - A Formal Description Technique Based on an Extended State Transition Model. ISO Draft Proposal - ISO/DP 9074.

Kinoshita, T., Sugawara, K. & Shiratori, N. (1988) Knowledge-Based Design Support System for Computer Communication System. IEEE Jnl on Selected Areas in Communications, 6(5), 850-861.

Laffey, T. J. (1986) Fault Diagnosis. IEEE Potentials, 5(3), 8-11.

Liu, D-M., D. & Pelz, D. A. (1988) I-Test : Integrated Testing Expert System for Trunks. IEEE Jnl on Selected Areas in Communications, 6(5), 800-804.

Macleish, K., J., Thiedke, S. & Vennergrund, D. (1986) Expert Systems in Central Office Switch Maintenence. IEEE Communications Magazine, 24(9), 26-33.

Macleish, K., J., Thiedke, S. & Vennergrund, D. (1987) Application of Artificial Intelligence in Telecommunications. GLOBECOM '87, IEEE/IECE Global Telecommunications Conference Record, Tokyo, Japan, 1, 329-333.

Mantelman, L. (1986) AI Carves Inroads : Network Design, Testing, and Management. Data Communications, 106-123.

Mathonet, R., Van Cotthem, H. & Vanryckeghem, L. (1987) DANTES : An Expert System for Real-Time Network Troubleshooting. Proceedings of IJCAI-87, Milano, Italy, 527-530.

McDermott, J. (1982) R1 : A Rule-Based Configurer of Computer Systems. Artificial Intelligence, 19, 39-88.

Milliken, R., Cruise, A. V., Ennis, R. L., Finkel, A. J., Hellerstein, J. L., Loeb, D. J., Klein, D. A., Masullo, M. J., Van Woerkom, H. M. & Waite, N. B. (1986) YES/MVS and the Automation of Operations for Large Computer Complexes. IBM Systems Journal, 25(2), 159-180.

Muralidhar, K. H. & Irish, B. W. (1988) MAPCON : An Expert System for Configuration of MAP Networks. IEEE Jnl on Selected Areas in Communications, 6(5), 869-873.

Peacocke, D. & Rabie, S. (1988) Knowledge-Based Maintenance in Networks. IEEE Jnl on Selected Areas in Communications, 6(5), 813-818.

Prerau, D. S., Lemmon, A. V., Gunderson, A. S. & Reinke, R. E. (1985a) A Multi-Paradigm Expert System For Maintenance of an Electronic Telephone Exchange. Proceedings of the 9th International Computer Software and Applications Conference (COMSAC), Chicago, US, 280-286.

Prerau, D. S., Gunderson, A. S., Reinke, R. E. & Goyal, S. K. (1985b) The COMPASS Expert System : Verification, Technology Transfer and Expansion, Proceedings of the Second Conference on Artificial Intelligence Applications, Miami Beach, FL, 597-602.

Rabie, S., Rau-Chaplin, A. & Shibahara, T. (1988) DAD : A Real-Time Expert System for Monitoring of Data Packet Networks. IEEE Network, 2(5), 29-34.

Shiratori, N., Takahashi, K. and Noguchi, S. (1986) IDESS/85 : Intelligent Support System For Protocols and Communication Software Development. International Conference on Communications ICC '86, Integrating the World Through Communications-Conference Record, Toronto, Canada, 1, 543-548.

Shortliffe, E. H. (1976) Computer-Based Medical Consultations : MYCIN, Elsevier, New York.

Sutter, M. T. (1986) The SMART Project : an Approach to Expert System Integration and Evaluation in the BOCs. International Conference on Communications ICC '86, Integrating the World Through Communications-Conference Record, Toronto, Canada, 2, 1230-1232.

Takahashi, K., Shiratori, N. & Noguchi, S. (1988) An Intelligent Support System for Protocol and Communication Software Development. IEEE Jnl on Selected Areas in Communications, 6(5), 842-849.

Thickett, E. (1986) Managing a Multi-Device Network. Networks 86, Proceedings of the European Computer Communications Conference, London, England, 57-62.

Vesonder, G. T., Salvatore, J., Stolfo, S. J., Zielinski, J. E., Miller, F. D. & Copp, D. H. (1983) ACE : An Expert System For Telephone Cable Maintenance. Proceedings of the IJCAI-83, Karlsruhe, W. Germany, 1, 116-121.

Welin, C. W. & Skagerwall, R. (1986) Design of An Expert System and Man-Machine Interface for the Operation and Maintenance of AXE Telephone Exchanges. Proceedings of the 1986 International Zurich Seminar on Digital Communications, New Directions in Switching and Networks, Zurich, Switzerland, 205-208.

Westcott, J., Burruss, J. & Begg, V. (1985) Automated Network Management. Technical Note, Bolt Beranek and Newman Inc., Cambridge, Massachusetts, USA.

Wright, J., R., Miller, F. D., Otto, G. V. E., Siegfried, E. M., Vesonder, G. T. & Zielinski, J. E. (1984) ACE : Going From Prototype to Product with an Expert System. ACM Annual Conference, 24-28.

Zeldin, P. E., Miller, F. D., Siegfried, E. M. & Wright, J. R. (1986) Knowledge Based Loop Maintenance Operations : The ACE System. International Conference on Communications ICC '86, Integrating the World Through Communications-Conference Record, Toronto, Canada, 2, 1241-1243.

Zhan, W-D., Thanawastien, S. & Delcambre, L. M. L. (1988) SIMNETMAN : An Expert Workststion for Designing Rule-Based Network Management Systems. IEEE Network, 2(5), 35-42.

Zhang, Y-X., Takahashi, K., Shiratori, N. & Noguchi, S. (1988) A Knowledge-Based System For Protocol Synthesis (KSPS). IEEE Jnl on Selected Areas in Communications, 6(5), 874-883.