Appeared in the Proceedings of the IJCAI95 Workshop on AI in Distributed Information Networks

Experience and Trends in AI for Network Monitoring and Diagnosis

Christopher Leckie
Artificial Intelligence Systems Section
Telstra Research Laboratories
770 Blackburn Road, Clayton
Victoria 3168, AUSTRALIA
c.leckie@trl.oz.au

Abstract

In this paper, we outline two practical Artificial Intelligence (AI) systems that we have developed for use in network monitoring and fault diagnosis. Based on this experience, we discuss the factors that were important to the success of these systems, and highlight our direction for future research in this field.

Introduction

This paper is divided into two parts. In the first part, we outline two applications where we have used AI techniques to solve practical problems in telecommunications network management. In particular, we focus on the tasks of performance monitoring and fault diagnosis. Based on this experience, we then highlight the key factors that affected the success of AI techniques for these tasks. In the second part of the paper, we describe the direction of our ongoing research in this area, which reflects our expectations of what will be required to develop these kinds of systems for future applications.

Lessons from two practical applications

Although there are many different functions involved in network management, three functions that are commonly associated with AI techniques are (1) performance monitoring, (2) fault diagnosis, and (3) network control and reconfiguration. These three functions form a natural hierarchy. At the lowest level, we detect abnormalities in the performance of our network. At the next level, we need to diagnose the cause of these abnormalities. At the highest level, having identified the root cause of the problem, we can then devise a suitable course of action to address the problem.

Each level involves some degree of expert analysis or problem-solving, and is thus a candidate for the application of AI techniques. However, we have found that in order to gain acceptance of AI systems by users, it is important to take an incremental approach by introducing this new technology into the lower levels first. Let us examine two practical applications that demonstrate this approach.

Monitoring and diagnosis

Our first application was designed to automate the tasks of performance monitoring and fault diagnosis of transmission equipment for a special purpose telephone network. Before the system was introduced, highly experienced staff were responsible for analysing on a daily basis large volumes of low-level performance statistics, such as traffic overflow and transmission quality. From these statistics, they would detect abnormalities, diagnose the cause of the problem, and develop a plan to rectify the problem. However, the sheer volume of data available (up to 3 MBytes each day) made it impossible for the experts to check every aspect of the network. Consequently, it was suggested that AI techniques be used to help reduce the workload.

Although AI could have been applied to each of the three levels in this application, it soon became clear that there were greater savings to be made in automating the lower-level, data intensive tasks of monitoring and preliminary fault diagnosis. This would then free the experts to concentrate on the more complex tasks of diagnosing more subtle faults and developing strategies to remedy the faults.

The architecture of our system is shown in Figure 1. The first requirement was to build a stable platform that could store the different types of data that would be analysed by the system. Of all the data collected each day, only a small proportion is likely to indicate a network fault. As a result, we developed a connectionist data filter that could quickly detect abnormalities in large volumes of raw data. This filter was designed to be easily customised by the experts, so they could verify and refine the knowledge used by its simple method of reasoning. This approach enabled the experts to gain confidence in the system using data with which they were familiar.

Once the knowledge in the data filter had stabilised, we were then able to introduce a rule-based expert system to perform more detailed diagnosis based on the output of the data filter. Whereas the data filter was designed to analyse

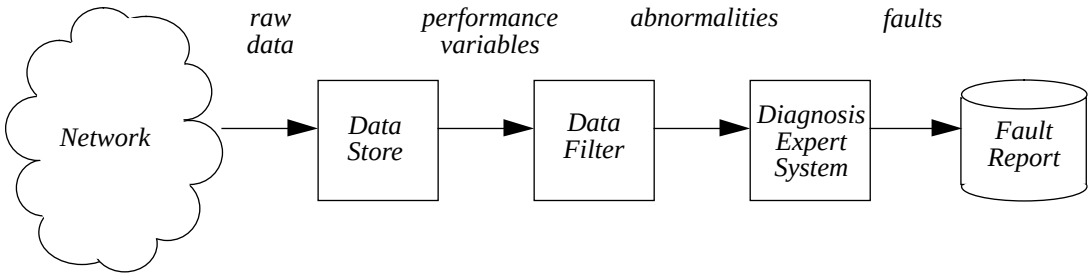

Figure 1: Architecture of the performance monitoring and fault diagnosis expert system

each network element in isolation, the diagnosis expert system can correlate abnormalities between several network elements. This enabled us to isolate sympathetic responses in network elements that are attached to a faulty element.

As the knowledge base of the expert system was refined, the experts gained sufficient confidence in the system to migrate from using the output of the filter to that of the expert system. This meant that they could concentrate on how to fix faults, rather than detecting and diagnosing faults. A more detailed description of this system appears in (de Beler et al. 1994).

Event correlation

Our second application is designed to correlate and prioritise events from a backbone data network. It is built on an existing platform that collects, stores and displays events from different layers of the data network. Examples of events include the status of front-end processors, point-to-point links, and X.25 virtual circuits. Several thousand events such as these may be generated each day.

Our role has been to develop a rule-based expert system that correlates related network events as they arrive. Based on these patterns of events, the system maintains a list of problems that have been detected in the network. In contrast with the previous application, which involved off-line analysis of averaged performance statistics, this application involves processing individual event records that arrive asynchronously in real-time.

At the heart of the system is a set of rules, where each rule defines a pattern of events that constitute a problem. Each rule defines a problem in terms of the type and number of events that must occur, a set of bindings between the fields of each event, and the maximum lifetime of the problem. When the arrival of an event triggers a new problem, subsequent events that arrive within the lifetime of the problem can be added if they match the corresponding rule. The system also incorporates heuristics to handle situations where several rules match the same event, or the lifetime of a problem expires before all the necessary events have been seen.

The main benefit of this system is to reduce the volume of data that the users need to analyse, thus enabling them to spend more time on tasks that require higher skill levels, such as diagnosing the cause of the problem, and fixing the fault. Although there is scope for using AI techniques for these latter tasks as well, it is important to gain the confidence of the users in this new technology first. By starting with a task that requires simpler reasoning, it is possible to develop a knowledge base that the users can quickly learn to verify and refine themselves. In this way, they develop a greater degree of ownership of the system. This system will be trialled in the second quarter of this year.

Key factors

Based on these two applications, we have been able to identify several key factors that contribute to the success of AI techniques in this area:

(1) Start with lower level, data intensive functions first, such as performance monitoring. Functions such as detailed diagnosis or control require more complex reasoning, making them riskier and harder for the experts to verify and trust.

(2) Gain user acceptance incrementally, by concentrating on a small set of commonly occurring problem types at first, rather than trying to analyse everything on the first attempt. This will reduce development time so that users can benefit from the system sooner.

(3) Make the system scalable and easily customised by the users. Often experts find it hard to articulate their knowledge until they have a practical tool to use.

(4) Target applications that already have a stable platform for data collection and storage. Without this, system developers are confronted with a “moving target”, and there is the additional cost of building multiple interfaces to data that is stored in different formats. It is also difficult to verify the results of knowledge acquisition because of the inaccessibility of test data.

Future Trends

As network managers grow more confident with AI techniques, they are willing to tackle more complex applications. Networks continue to grow larger and more complex, thus increasing the volume of data that needs to be analysed in order to monitor network performance. Furthermore, modern network elements are able to report more detailed performance data than was previously possible. As the capacity of data collection platforms also increases, it will become feasible to access lower-level alarms and individual call failure records. This will make it possible to diagnose more subtle problems, and help to identify faults earlier before their effects become widespread.

However, detailed analysis of large volumes of low-level data in real-time will push existing AI systems well beyond their limits. Consequently, there will be a growing demand in the future for distributed AI (DAI) systems that can divide the processing requirements between multiple systems. This approach also has the potential to reduce the volume of management data that needs to be passed between sites, as well as reducing the effects of any single system failing.

While considerable attention has been paid to DAI for control and reconfiguration (Weihmayer and Tan 1992), we believe that there are important opportunities in the areas of monitoring and low-level diagnosis. In this context, the aim is to localise the analysis of detailed monitoring data wherever possible. However, we still need to exchange information between systems to correlate evidence for suspected problems. If too much information is exchanged, the advantages of distribution will be lost, while if too little information is exchanged, it will be difficult to identify low-level problems. A key issue for research is to explore this trade-off between efficiency and accuracy.

Our ongoing research in this area aims to build on our experience from the applications we described earlier. By building on existing systems, our aim is to create a migration path for introducing DAI into practical network applications. Consequently, we expect that users are likely to be more responsive to this new technology.

Acknowledgements

The applications described in this paper are result of research and development conducted by members of the AI Systems Section, Telstra Research Laboratories. The permission of the Director of Research, Telstra, to publish this material is hereby acknowledged.

References

De Beler, M., Kowalczyk, A., Leckie, C., Rowles, C., Sember, P., and Senjen, R. 1994. AI for Managing Telecommunication Networks. In *Worldwide Intelligent Systems: Approaches to Telecommunications and Network Management*, Liebowitz, J. and Prerau, D.S. (eds), IOS Press, Amsterdam, ISBN 90 5199 183 5, August 1994, pp. 87-108.

Weihmayer, R. and Tan, M. 1992. Modeling Cooperative Agents for Customer Network Control Using Planning and Agent-Oriented Programming. In Proceedings of Globecom'92, pp. 537-543.