View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by NASA Technical Reports Server

N88-29385

Intelligent Resource Management for Local Area Networks: Approach and Evolution[1]

Roger Meike
Martin Marietta Denver Aerospace
Space Station Program
P.O. Box 179 (MS D1744)
Denver, Co. 80201

Abstract

The Data Managment System network is a complex and important part of manned space platforms. Its efficient operation is vital to crew, subsystems and experiments. AI is being considered to aid in the initial design of the network and to augment the management of its operation. The Intelligent Resource Management for Local Area Networks (IRMA-LAN) project is concerned with the application of AI techniques to network configuration and managment. A network simulation was constructed employing real-time process scheduling for realistic loads, and utilizing the IEEE 802.4 token passing scheme. This simulation is an integral part of the construction of the IRMA-LAN system. From it, a causal model is being constructed for use in prediction and deep reasoning about the system configuration. An AI network design advisor is being added to help in the design of an efficient network. The AI portion of the system is planned to evolve into a dynamic network management aid. This paper describes the approach, the integrated simulation, project evolution, and some initial results.

The Intelligent Resource Managment for Local Area Networks (IRMA-LAN) project is concerned with the application of Artificial Intelligence (AI)techniques to network management. As part of this project, a prototype AI subsystem is being developed in Lisp on a Symbolic 3675 computer to test and demonstrate several of the concepts central to this project. By implementing an evolutionary design schedule, a rapid prototype was quickly assembled and then augmented. This allowed us to have a demonstration of some basic concepts very quickly. As the AI subsystem develops, more complexity is added to it. There are several benefits of developing an AI subsystem in parallel with the application system. These benefits are expected to generalize to other AI subsystems.

Our initial studies identified eight candidate areas for the application of AI techniques to network configuration and management.[2] The areas identified include fault detection, fault diagnosis, fault recovery, fault prevention and fault prediction as part of fault tolerance, and process addition, process deletion and optimization as part of dynamic reconfiguration. Resource evaluation and configuration analysis are important shared tasks within these areas. A key discovery was that these tasks are important to the design of networks as well as to their operation. The ability to evaluate and use available resources efficiently is central to both a good design and smooth operation. This commonality allowed us to develop some of the concepts central to a dynamic network manager in a network design aid. The process we used is described below.

Problems

Currently, no human has the job to manage computer networks in the same way that we hoped the IRMA-LAN system would (i.e. dynamically, while the network operates). There are no real experts, thus, we had to first invent the job and then find out how to be good at it. There *are* people whose jobs are related to network management, and thus would have more expertise and insight into what this hypothetical network manager should do. We used knowledge and basic logic of these pseudo-experts to develop the rules of the AI subsystem.

Some basic problems that can arise when building expert systems are that: (1) experts sometimes lose interest in the your system, (2) don't understand its design, or (3) consider it a very low priority because they expect little gain from it.[3] We became interested in these problems when we decided to build a prototype IRMA-LAN system but had no experts to assist us. To create a true expert system we needed experts who were interested enough to give their expertise in return for immediate benefits. The way in which the project evolves is critical to keeping the interest of the experts. If there is an immediately useful by-product, experts are much more likely to be interested in helping to develop the AI subsystem. Also, by involving the experts in the evolution, they have a much better understanding of the resulting AI subsystem. Our approach was to make extensive use of our rapid prototyping environment to build a subsystem that did not incorporate expertise, but that would be immediately useful to an expert and could evolve to have the expertise of the network management tool originally desired. A design aid was chosen for implementation because it could both help the expert in the network design and develop into the dynamic manager.

The Design

Although the goal of the IRMA-LAN project is to assess the role of AI techniques in network managment, the prototype AI subsystem must demonstrate those AI techniques as well as the environment in which they operate. As much effort is put into the conventional software as is put into the AI portions of the subsystem. In most cases, the user does not know or need to know whether a specific function is implemented using AI or conventional programming techniques. An emphasis of this project is to make a functional subsystem rather than one to demonstrate AI concepts. Integration is very important to this effort. The qualitative model, quantitative model and expert analysis portions interact with each other continuously in IRMA-LAN. The user interface presents these interacting portions as a single subsystem. Parallel interdependent development ensures that the portions of our subsystem will be well integrated. The various interactions of a complete IRMA-LAN system are shown in figure 1. From the start of this project the simulation model, user interface and expert analysis portions of the subsystem have been developing. Early in the development a network interface was added. Thus, all portions of the subsystem are now being developed in parallel.

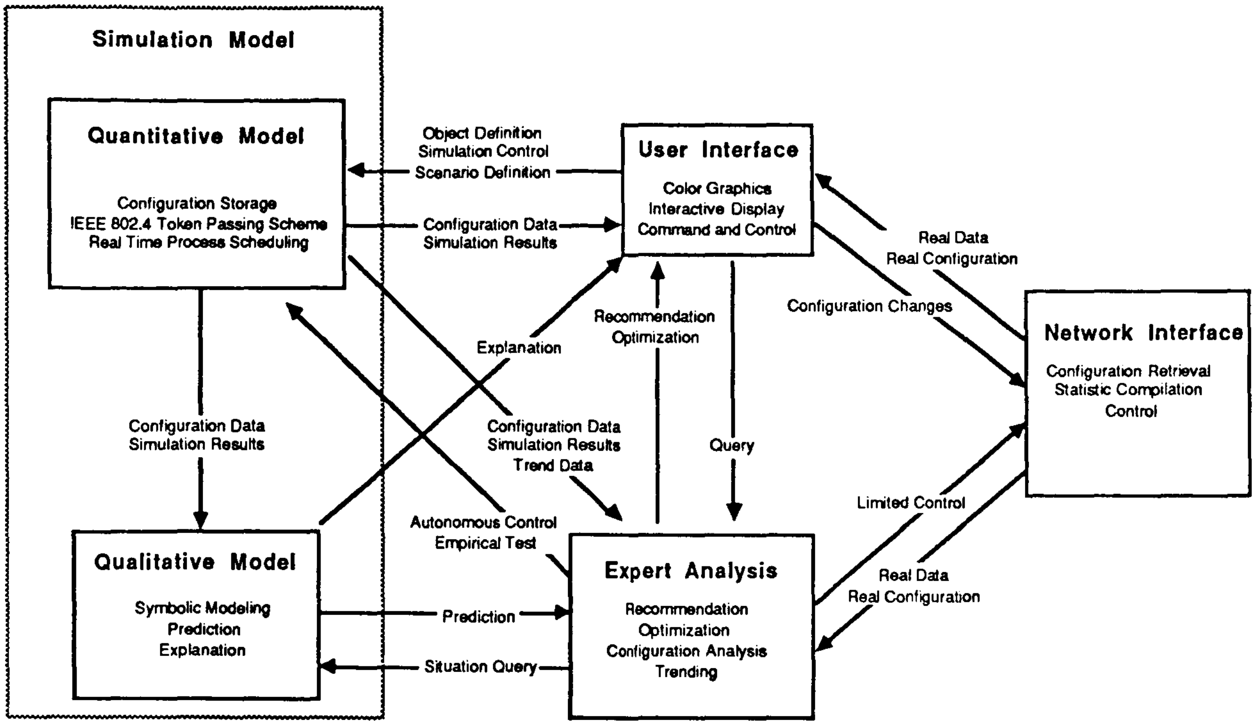

Figure 1 - Final phase components and interaction

Over four phases we plan to evolve our knowledge base from a straight forward expert system for network design to a dynamic network management tool. These phases parallel the expertise that we are receiving. Our network management experts are people who design networks and network management programs, not people who manage networks themselves. Therefore the most direct use of their knowledge was to build a design aid. In this case, the AI subsystem is an advisor to a network designer. By abstracting this information to a higher level, we are building a knowledge base of principles involved in the network management problem. From these basic principles we are deriving a dynamic network management aid which will help in real-time network management.

In the first phase, the subsystem was viewed as a network design aid. Network designers deal with large amounts of information. Our application benefits them by presenting these large amounts of information in a clear and concise manner. Using rapid protoyping techniques, we designed a very flexible system which allows the designer to change configurations easily. The first phase design consisted of a computer network simulation model and a preliminary user interface. Network designers could then test new designs and get immediate feedback through color graphics. The simulation model is flexible enough to allow easy changes, yet complete enough to allow reliable feedback to the user. This system is simple, but it still provides an improvement over the pencil and paper techniques that network designers typically use. Once this system was shown to the designers (potential users), they became eager to provide suggestions to improve it. Because of the rapid prototyping environment, we were able to implement their suggestions quickly and keep their interest.

The second phase involves connecting the system to a network so the designer can interact with a real network as well as the simulated one. In addition to the simulation model and user interface, a network interface is added. This allows for much more detailed designs and tests on real hardware, while maintaining flexibility through the simulation model. We are currently entering this phase of our project.

During phases one and two, the AI portion of the project begins. Expert analysis rules and a qualitative model are developed in parallel with all the phases of the project. As we add functionality to the whole system, we add AI functions as well as conventional ones. The expert analysis rule base is created using vertical slice[4] techniques in which a particular function is explored in detail and integrated into the system. This entails finding a suitable problem that a designer may want to work on and discovering the rules that govern the decision-making process for that particular problem. Using this method, demonstrations of AI potential can be rapidly developed. Some of the reactions to the initial configuration of the system led to the development of a set of fault diagnosis rules that set the framework for the knowledge base and are still a major part of our demonstrations.

Because of the lack of true expertise in this limited domain, a qualitative model is being constructed to allow for deep reasoning that will fill in the gaps in the knowledge base. This portion of the simulation model is being developed from specific cases and logical analysis of the quantitative model.

The primary goal of the AI work during phases one and two is to develop an advisor. This means that the IRMA-LAN system is to be used as a network design tool in which the designer uses the AI portions of the system to help in various portions of the design. In the next two phases this thrust changes. The IRMA-LAN system makes a transition from design tool to management aid. The AI subsystem makes a transition from advisor to actor. The IRMA-LAN system becomes a part of the network management system and reacts to situations in the network rather than queries from a user. Based on these decisions it takes the appropriate actions.

In the third phase the IRMA-LAN system recommends an initial configuration of the network based on various actions of the designer. The system is still used as a design aid, but now it has taken on a more active role. Based on scenarios that the designer runs, and other information it gathers during the design process, it takes an active role in initializing the network for use through the network interface.

In the final phase, the IRMA-LAN prototype acts as a manager which reacts to the network and dynamically interacts with it to ensure efficient operation. This may include allocating processes to processors, *tuning* timers and buffer sizes, reacting to predicted changes in performance, and other active functions. The AI system will have made the complete change from advisor to actor.

ORIGINAL PAGE IS
OF POOR QUALITY

Development Stage	AI Function	AI Development	System Function
Phase 1	Advisor	Rapid Prototype Vertical Slice	Design Aid with Simulation Model
Phase 2		Integrate Causal Model	Design Aid with Interface
Phase 3	Actor	Refine Rule Base	Initialization
Phase 4		Real Time	Dynamic Management

Figure 2 - Four phase design

Analysis

The design schedule presented here (figure 2) appears to be applicable to AI subsystems in many different domains. To achieve a truely integrated subsystem, it is a good idea to be involved with the design of the conventional system which the AI subsystem is to augment. Because of the rapid prototyping environments commonly used in AI development, it is possible to help in the design of the whole system rather than just the AI portions. Providing such tools for designers can result in many advantages. First, it allows feedback from experts throughout the design of the system. Experts are actively involved in the design of the AI portion of the system by reacting to it as it develops. Similarly, the AI system can react to and become part of the design. Second, the schedule provides demonstration capabilities very early on in the development cycle. This provides visibility that can assist in integrating AI into the system. Third, it provides a base for testing and growth. The more complete the environment is for testing code, the more reliable the tests are. A tool that is used for design purposes will be well tested by the users and will become a good test ground for new code. Fourth, the knowledge acquired in the design phase may be directly applied to the final system. In many cases, the principles involved in designing a system are central to its operation. AI functions that perform tasks such as abstracting out the causes of certain actions in the simulation model may be just as applicable to monitoring performance in a real system. Fifth, in the case where expertise is not readily available, a qualitative model may help. When there are no specific rules of thumb, a more general qualitative model can help the reasoning process and make for a more complete solution. Knowledge gained in developing a *quantitative* model was found to be very useful in the design of the *qualitative* model that we are currently working on for IRMA-LAN.

Notes

[1] The IRMA-LAN project is supported by Martin Marietta Denver Aerospace, Independent Research and Development task D-47S

[2] Meike, R. C., "The Application of Artificial Intelligence Techniques to Network Reconfiguration for Space Station", Proceedings: Artificial Intelligence - From Outer Space Down to Earth, October 1985, University of Alabama, Huntsville

[3] Some of the problems with dealing with experts are presented in
Bailey, P. A., and Doehr, B. B., "Knowledge Acquisition and Rapid Prototyping of an Expert System: Dealing With 'Real World' Problems", Proceedings: Conference on Artificial Intelligence for Space Applications, Huntsville, Alabama, November, 1986.

[4] Vertical slicing technique refers to studying a particular problem in depth, thus taking a vertical slice of the problem domain. The design aid itself is a horizontal slice of the whole system under development, but the knowledge base (at least at first) is developed one problem at a time. For more information on vertical slicing see
Gates, K. H., Adelman, L., and Lemmer, J. F., "Management of AI System Software Development for Military Decision Aids" IEEE Symposium on Expert Systems and Government, Washington D.C., 1985.

ORIGINAL PAGE IS
OF POOR QUALITY