AN OPENVIEW BASED ATM NETWORK MANAGEMENT SYSTEM

NIGEL COOK, ROBERT COOTE, DAVID HORTON, GEOFF THOMPSON, AND HIROSHI SUZUKI[1]

***Abstract*—Asynchronous Transfer Mode (ATM) networks are emerging as a major broadband networking technology. The management of these networks creates new challenges for both private network operators and public telecommunications service provider communities due to the heterogeneous mix of ATM Switch equipment, and the need to establish, control and monitor end-to-end connections (virtual circuits) through a network. Considerable effort is being applied to the formulation of standards in this area, in particular by the ATM Forum organisation.**

CiTR has been working in association with NEC on implementing a prototype ATM Network Management System (NMS) using OpenView and SNMPv1 to control and monitor ATM Networks via the IETF AToMMIB. This international collaboration between teams based in Australia and Japan has developed successive phased releases of NMS software. The initial development focused on connection oriented OpenView map navigation and a graphically driven connection establishment interface. The latest release includes ATM autodiscovery, automatic optimal route selection when establishing Permanent Virtual Circuits (PVCs), and the subsequent monitoring and unattended restoration of the PVCs.

The paper summarises aspects of the development of ATM Network management software using the OpenView platform, and includes a description of the phased development strategy as well as the features and pitfalls of using OpenView in this application.

***Source of Publication*—OpenView Forum, Seattle, USA, June 1995.**

1 PERSPECTIVES

1.1 INTRODUCTION

Asynchronous Transfer Mode (ATM) is a high-speed networking technology. ATM’s speed, low latency and ability to handle all types of traffic over a single network make it ideal for a range of bandwidth-intensive applications, including multimedia, medical imaging, bank-check imaging, seismology studies, supercomputer applications, CAD/CAM, and more.

One factor influencing the pace of ATM adoption is standards, many of which are being formulated by the ATM Forum.

One difference between ATM and other networking technologies is the concept of “virtual circuits”. Negotiating a virtual circuit with nominated bandwidth and quality of service parameters is a feature of ATM circuit provisioning. Standards defined by the ATM Forum and IETF allow users to change network design and re-route traffic by manipulating virtual circuits. However, in many implementations to date this has been a manual process.

1.2 ATM FORUM

ATM Forum was founded in 1991 as an association of over 700 members to promote ATM standards, and to help bring ATM to the market quickly.

[1] Dr. Hiroshi Suzuki, NEC Network Research Laboratories, C&C Research Laboratories, 1-1,Miyazaki 4-Chome, Miyamae-Ku, Kawasaki, Kanagawa 216, Japan, e-mail: hiroshi@nwk.cl.nec.co.jp, phone: +81 44 856 2123, fax: +81 44 856 2230

The ATM Forum has devised a ATM network management model as shown in Figure 1.

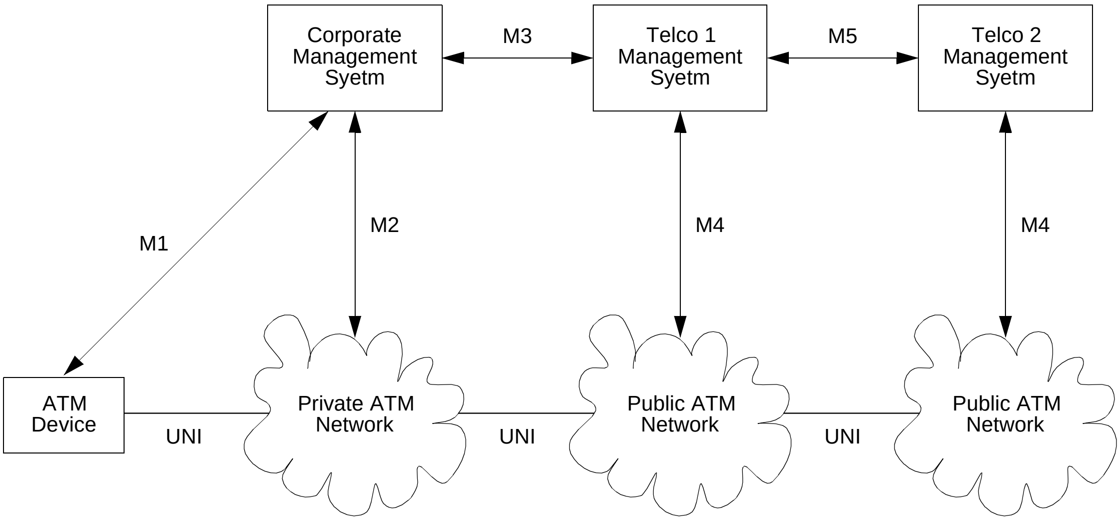

Figure 1: ATM Network Management Model

The ATM network management model defines five work areas for network management:

- Management of ATM end devices (M1)
- Management of private ATM networks or switches (M2)
- Management links between public and private networks (M3)
- Management of public networks (M4)
- Management links between 2 public networks (M5)

AToMMIB is an SNMP version 2 MIB developed by the IETF (Internet Engineering Task Force) for ATM management. After a year (and 8 revisions) of development, it was released in August 1994 as RFC 1695. It supports semantics for the monitoring and setup of virtual circuits. Work done by co-authors at NEC includes ATM Forum submissions for definitions necessary for ATM auto-discovery support [1].

The OpenView network management system described in this paper supports M1 and M2 style management interfaces. The SNMP agent, which was also developed in this project, uses information from the UNI ILMI for reporting to management functions through AToMMIB.

1.3 OPENVIEW

OpenView was chosen as the development platform for this project. Published independent studies of market share and of available applications for various network management development platforms confirmed the market place acceptance of OpenView as an “Open Platform”. A summary of historical and projected figures for various platform alternatives is shown in figure 2[1] .

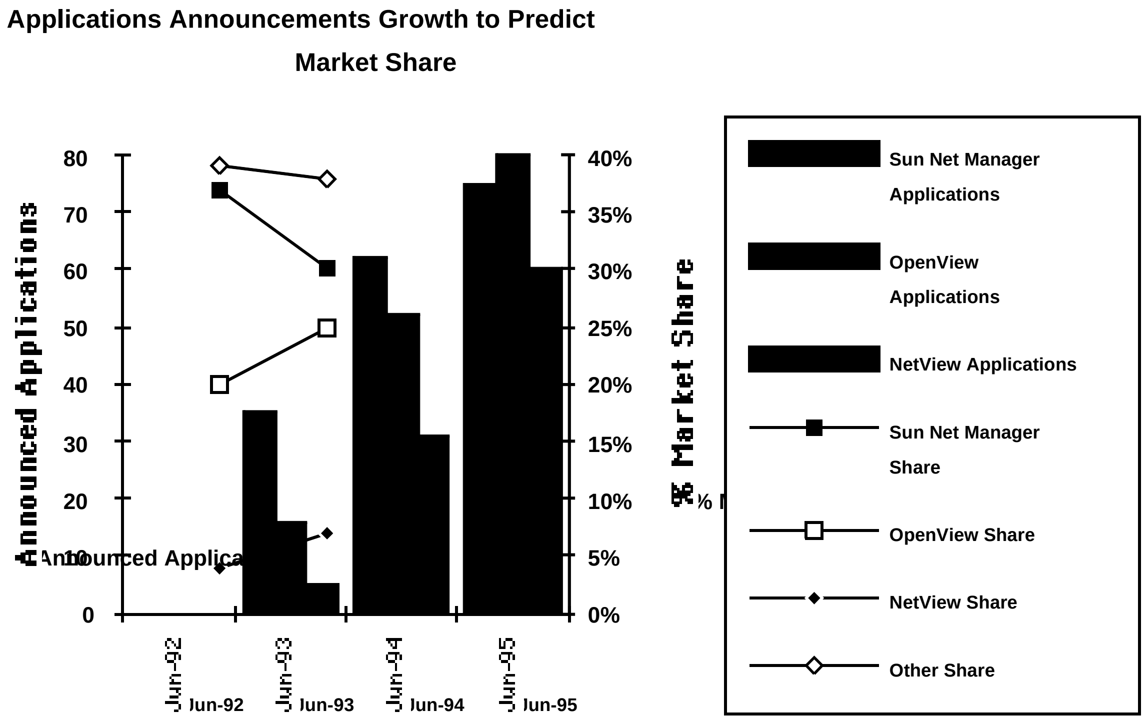

Figure 2: OpenView Market Performance

1.4 NEC/CITR DEVELOPMENT PROGRAM

The NEC/CiTR joint development program began in mid-1993. The program concentrates on implementing proof-of-concept network management prototypes based on open standards. The proof-of-concept software developed by CiTR is provided to NEC as an input to the further commercial product development activities. The NEC development program has resulted in the NEC ATOMVIEW product. This paper concentrates on the proof-of-concept development program only.

The early adoption and use of recommendations and drafts from bodies such as ATM Forum and the IETF is an important component of the development project, and provides valuable experience for later product implementations.

The development program is structured as a series of phased developments, with a number of functionality releases in each phase. In later sections of this paper the functionality defined in these phases is described.

2 PHASE 1 DEVELOPMENT PROJECT

2.1 FUNCTIONALITY

The Phase 1 development project built a basic ATM network management system that allowed monitoring and manually initiated control of a set of ATM hosts and switches using SNMPv1. Capabilities included:

- PVC connection management
  - directly allowing IP access translation
  - based on AToMMIB and proprietary end point IP to VCI MIB
- Multiple views of connections, both logical, and physical
- Drill down connections and nodes

[1] Third Party Applications data is sourced from Managing Distributes System, Jill Huntington-Lee, Brandywine Network Associates, published by Faulkner Information Services (Pennsauken,N.J.). Platform Market share data is from IDC,1994 data based on unit shipments, rather than revenue.

- Context sensitive object detail reports
- Fault management
- SNMP V1/V2 bi-lingual agents for both host and switch.

CiTR in-house tools allow automated conversion from SNMP MIB specifications to software agents. This allowed easy tracking of evolving IETF specifications.

2.2 ARCHITECTURE

The architecture (shown in Figure 3) consists of some OpenView based applications, and supporting reporting tools. These reporting tools are a combination of the OpenView object database (OVwDb) applications and some SNMP applications, as well as atm_lan, which is an OVw integrated application, performing manipulation of the OpenView Windows (OVw) map, symbols, and OVwDb.

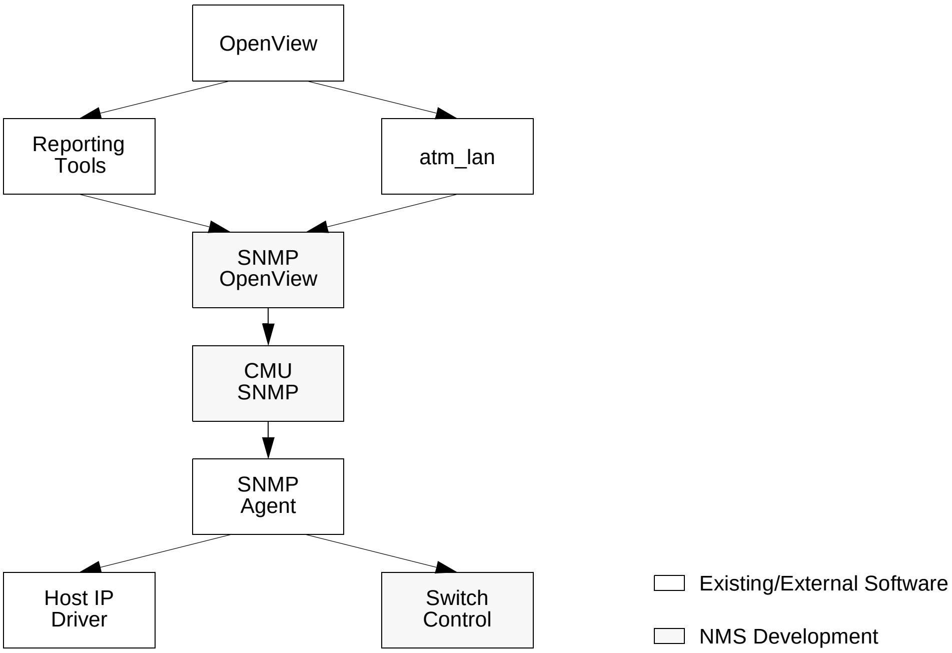

Figure 3: Phase 1 High Level Architecture

2.3 USER INTERFACE

Figure 4 shows a sample interface screen from the application.

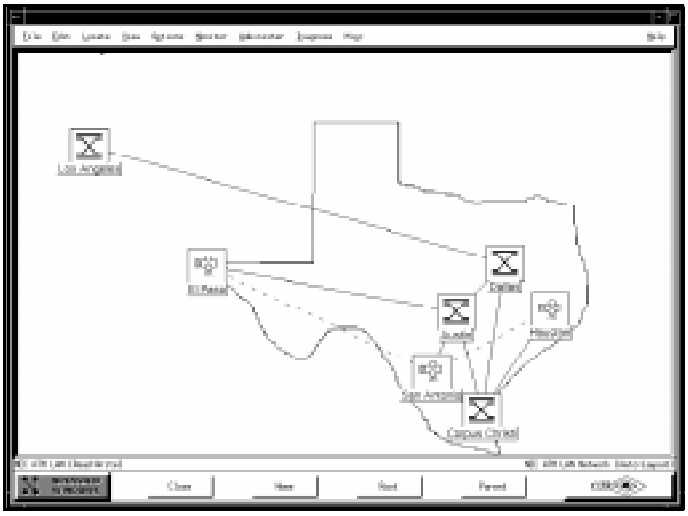

Figure 4: Sample Interface Window

The map shows the ATM switches and hosts. Physical connections between nodes are shown as solid lines. Logical connections (virtual paths/circuits) are shown as dashed lines.

3 PHASE II DEVELOPMENT PROJECT

3.1 FUNCTIONALITY

The Phase II Development concentrated on the addition of automation functions to the basic ATM NMS established during Phase 1. The following functions were added:

- Fault & topology monitoring
- SNMP based ATM auto-discovery (automatic physical connectivity) using AToMMIB (RFC 1695 Final)
- Virtual Circuit Tracing
- Automatic path selection, using shortest path (fewest hops) algorithm as sample heuristic
- Unattended automatic PVC re-routing

3.2 ARCHITECTURE

The phase 2 development involved a series of enhancements to the phase 1 software. Another integrated OVw application "atm_mon" was added, as shown in Figure 5. This application contains functions that detect nodes and ports transitioning to the "down" state. This is achieved by both actively polling status, and passively catching and processing "traps" which report the change of state. The application split exists to ensure maximum responsiveness in atm_lan, as atm_mon can suffer from delays due to SNMP timeouts in the processing loop.

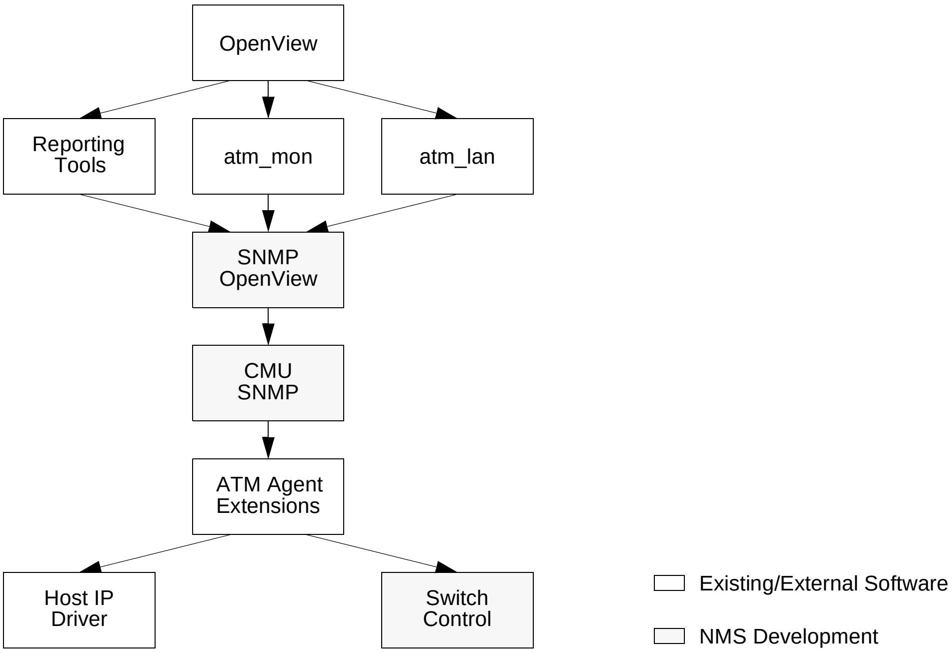

Figure 5: Phase 2 High Level Architecture.

The atm_lan application was further enhanced in the phase 2 development. Association between major functional blocks was achieved with a combination of internal APIs, and use of OpenView OVw callback registration, OVw actions and Xt Intrinsics timers. (See Figure 6).

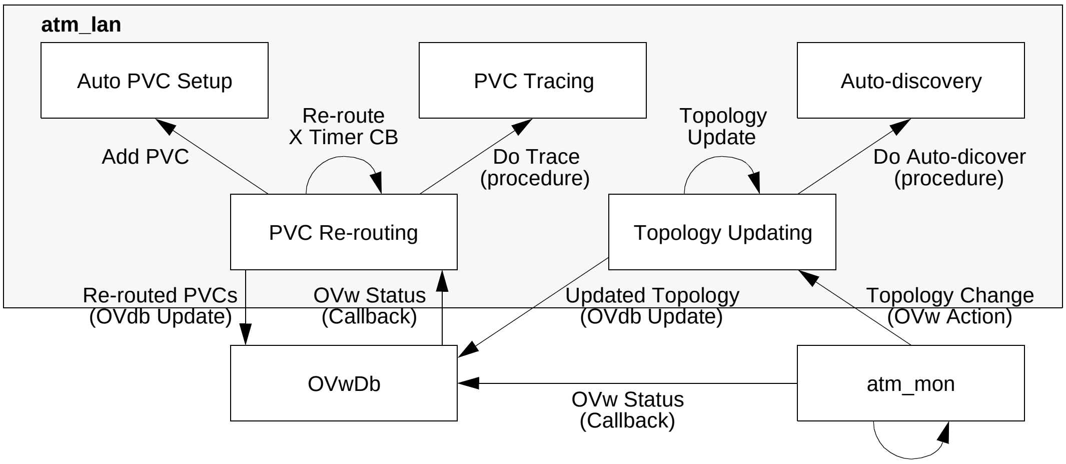

Figure 6: ATM_LAN/ATM_MON Interaction

3.3 USER INTERFACE

Figure 7 shows the OpenView IP submap window, with the icons representing the ATM submap on this page. Clicking on the ATM submap icon (nwkatm-cl-nec) explodes into the top level ATM submap (as in Figure 4), which is populated by the ATM auto-discovery process.

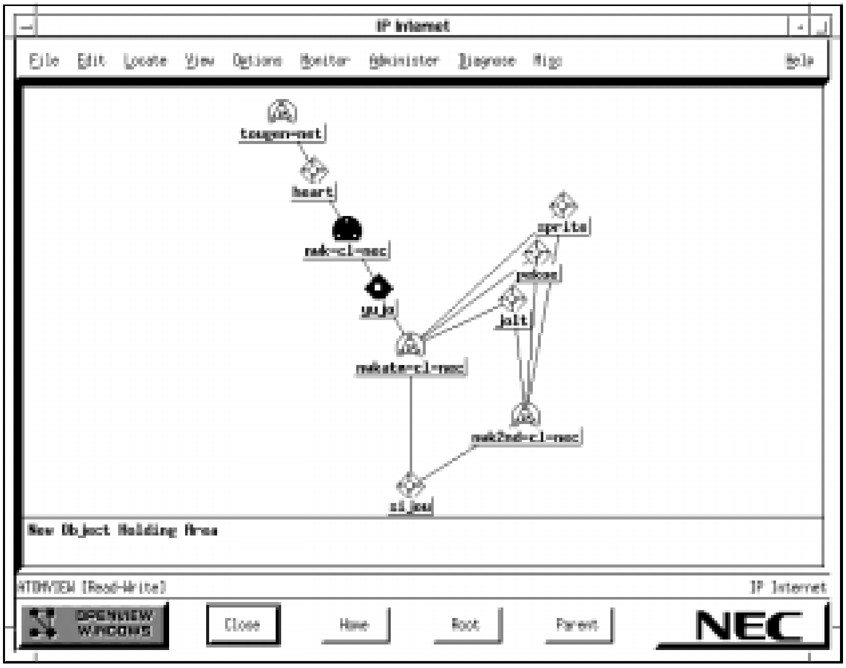

Figure 7: ATM Autodiscovery Integration

4 ATM AUTO-DISCOVERY

The addition of the ATM auto-discovery functionality is a key feature in the provision of a network management system in a private LAN-centered ATM networking environment.

The ATM Auto-discovery functionality is important for

- initially determining the physical connectivity of the network, and
- on-going monitoring of the topology.

Other elements of functionality, such as the unattended PVC re-routing, rely on the automatic invocation of auto-discovery to maintain a view of the network topology on detection of port status change. The OpenView object status change callback triggers processing on detection of a port operational status change. Subsequent processing by atm_mon using information obtained from neighbour information fields defined in RFC1695 determines topology change, and then an OpenView action triggers invocation of auto-discovery to resolve the updated network configuration.

The auto-discovery algorithm minimises the host traffic as it is linear to the number of nodes in the system. A sample of the timings and performance of the auto-discovery algorithm (based on a Sun Sparc 10) is presented in Table 1, and Figure 8.

Number of Nodes	Number of Connections	Time to Auto-discover
4	3	27.33s
4	6	29.00s
8	7	55.6s
8	28	75.25s

Table 1: ATM Autodiscovery Timings

Only a small percentage of the time required for autodiscovery is associated with network accesses. The majority of processing is related to OpenView database manipulation.

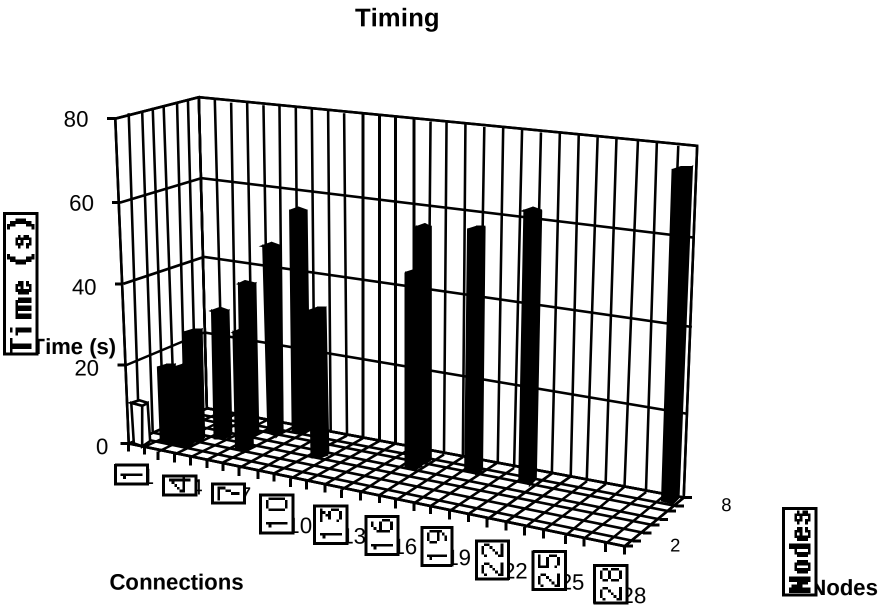

Figure 8: Auto-discovery Performance Measurement

5 OPENVIEW DEVELOPMENT PROBLEMS

OpenView is an effective development environment, but suffers in a number of areas that affects the development effort and the clarity of presentation to the end user. In particular,

- Lack of transaction functionality with which to group multiple object add/delete operations.

In the application, there are long sequences of operations, involving the OpenView databases, MIBs on a number of different nodes, plus an external database. During an involved creation process, the OpenView model requires we keep separate records of components created, so that if we have to rollback we can delete the components that were created prior to the rollback. Rolling back a delete is also a complicated process. These limitations force the application to do up front checking that operations are likely to succeed before sending requests.

- the OpenView palette of icons and connections allows no program control or hiding.
  This causes the internal symbols in the system to be inappropriately exposed to the user as a possible choice
- connections are poorly supported as follows:
  - locating connections for an object
  - there is a limitation of only one connection palette
- Separation of OVwDb from mapdb
  The management system included a status monitoring daemon which updated the status on objects which were associated with the MIB only, i.e. did not relate to symbols on a submap. However, due to OpenView limitations, a separate OpenView session was necessary to provide a submap in order to set status on objects.
  Another part of the management system included a background recovery application that re-routed PVCs and detected network topology changes. In order to make these changes to the map, a read/write connection to a map was required which would interact unfavorably with operator usage of the map. While it may have been possible to batch up symbol and submap changes until there was a writeable map available, this introduced considerably more application complexity.
- Program versus User Control distinction poor
  Programatically cannot distinguish between user versus software initiated actions.
- Performance Problems
  Operations that update the OpenView database are very slow. Even with the relatively small test network, the majority of elapsed time for the operation, such as automatic re-routing, is spent updating the OpenView representations and database rather than manipulating the actual network.

6 FUTURE DIRECTIONS

- Improved host MIB definitions
- World-Coordinate (geographic) based display

7 CONCLUSIONS

OpenView platform was an adequate development tool for the Broadband Network Management applications, but has severe limitations. The use of automated agent generation tools allowed easy tracking of standards. The project demonstrates the possibility of an OpenView based Broadband Network Management system capable of controlling networks containing multiple vendor switches, based upon the open standards defined by the IETF and ATM Forum.

REFERENCES

[1] Suzuki, H., McCloghrie, K. “PNNI & ILMI Requirements for Auto ATM Discovery by SNMP-based NMS” ATM Forum/94-0499 May 10-13 1994.

[2] Tesink, K., “(Re)Configuration of ATM Virtual Connections with SNMP”, The Simple Times Aug. 1994. Dover Beach Consulting, Inc. Mountain View, CA.

[3] Thyfault, M.E. “The Future is Spelled ATM -- Speed and the ability to handle all types of traffic over a single network makes asynchronous transfer mode a hot networking technology” INFORMATIONWEEK Issue 508, p 50. CMP Publications Inc.