We have also described our initial prototype of this architecture, which was deployed in a two-level hierarchy to manage a network of ATM equipment.

The main feature of our architecture is the generic network model, based on ITU-T Recommendation G.805. The software components used to implement this architecture can be deployed on management clusters, situated within a hierarchical structure based upon geographical separation. CORBA is the technology of choice for the distributed communication between the components, though we also offer a north-facing OSI interface at each level in the hierarchy.

A system supporting this architecture will provide a single solution for service providers in their multi-technology, multi-vendor networks.

REFERENCES

[1] ITU-T Recommendation G.805 (11/95), Generic Functional Architecture of Transport Networks

[2] European Telecommunications Standards Institute (ETSI), I-ETS 300 653: Telecommunication Management Network (TMN)—Generic Managed Object Class Library for the Network Level View, March 1996.

[3] ISO/IEC DIS 10165-4, ITU-T Recommendation X.722, Information Technology—Open Systems Interconnection—Structure of Management Information—Part 4: Guidelines for the Definition of Managed Objects (GDMO), September 1991.

[4] OMG, CORBA: Common Object Request Broker Architecture, OMG, August 1995.

[5] Gordon McNair and Jason Etheridge, *Developing Agents with the Managed Object Toolkit,* Proceedings of the International OpenView Forum Conference, Anaheim, USA, June 1997.

[6] ITU-T Recommendation M.3010—Principles for a telecommunications management network.

[7] ITU-T Recommendation M.3100—Generic network information model.

[8] G. Chen and Q. Kong, *Integrated TMN Service Provisioning and Management Environment,* in *Integrated management in a Virtual World,* Proceedings of the 5th IFIP/IEEE International Symposium on Integrated Network Management, A Lazar, P. Saracco and R. Stadler (eds.), Chapman & Hall, 1997.

[9] G. Chen, M. Neville and Q. Kong, *Distributed Network Management Using CORBA/TMN,* Proceedings of the 7th IFIP/IEEE International Workshop on Distributed Systems Operation and Management (DSOM), L'Aquila, Italy, October 1996.

[10] G. Pavlou, K. McCarthy, S. Bhatti and G. Knight, The OSIMIS Platform: Making OSI Management Simple, *Integrated Network Management IV,* Chapman & Hall, Adarshpal S. Sethi, Yves Raynaud and Fabienne Faure-Vincent (eds.), Proceedings of the Fourth International Symposium on Integrated Network Management (ISINM), 1995.

[11] Makoto Nishio, Kazuo Takagi, Shinobu Sasaki, Satoshi Hasegawa, *ATM Flexible Access Node for Subscriber Feeder Networks*, IEEE GLOBECOM 97, pp883-888, 1997. Nov.

7 EVENT PROPAGATION

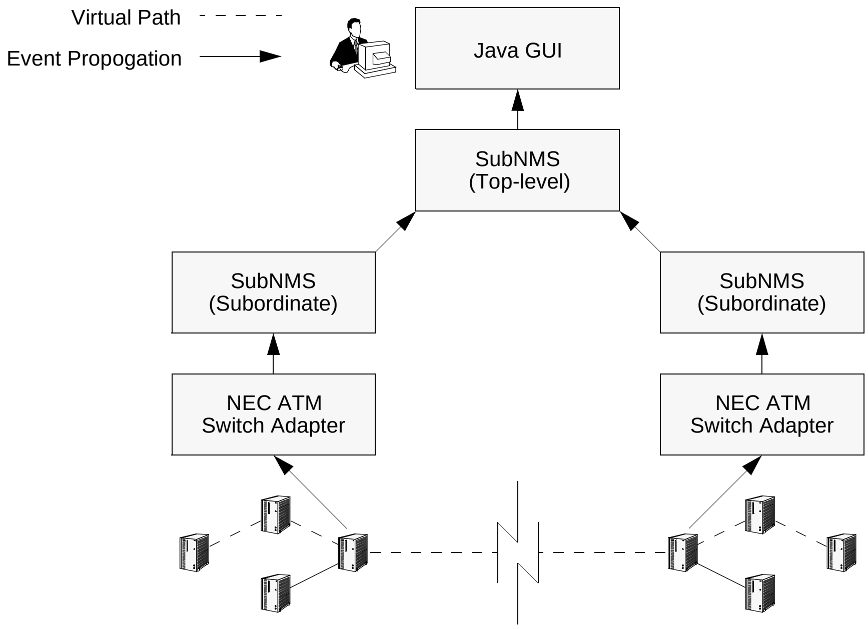

Figure 7: Event Propagation Using the Prototype NMS

Where the end-to-end VP establishment showed that top-down management could be accomplished, it was also necessary to demonstrate that bottom-up information flow would also be effectively propagated upwards.

After establishing a number of VPs through the top-level SubNMS, a fault was introduced into the system by disconnecting the fibre cable connecting two of the ATM network elements. This physical connection connected the two ATM sub-networks being managed, and was used by several of the VPs.

The failure of the physical connection was detected by the ATM network elements at both ends of the cable. The alarms raised were passed onto their respective OSIMIS/ISODE agents, which emitted CMIP M-Event-Reports indicating the problem.

These M-Event-Reports were received by both NEC ATM Switch Adapters, and was translated immediately into a CORBA event that was passed through the Adapters' IDL interfaces upwards to both subordinate SubNMS systems. (Note that due to the closely coupled nature of the Adapters and SubNMS systems, the events were propagated directly without making use of an OMG COSS Event Service event channel.)

When the subordinate SubNMS systems received the events, they were applied to the network model that each SubNMS maintains. This resulted in a number of the objects contained within the generic network model—particularly those representing connections affected by the faulted physical connection—changing state. This caused events to be emitted from the domain-independent interface supported by the NMS Core. The OSI agent and CORBA server components then caught these events, which were translated into M-Event-Reports and OMG COSS Event Service events respectively.

The CORBA events, transmitted through an event channel (one event channel being used per SubNMS) were received by the top-level SubNMS. These changes were applied to its own generic network model, which likewise caused events to be pushed upwards and emitted through its own OSI agent and CORBA server. In this way, the objects representing the VPs passing over the faulted physical connection in the top-level SubNMS become alarmed. This was displayed in the Java GUI associated with the top-level SubNMS when it received the CORBA event indicating the object had changed.

8 CONCLUSION

In this paper, we have presented an architecture for a distributed, hierarchical network management system, which is capable of managing network elements and management domains consisting of different technologies from different vendors.

6 END-TO-END PATH SETUP

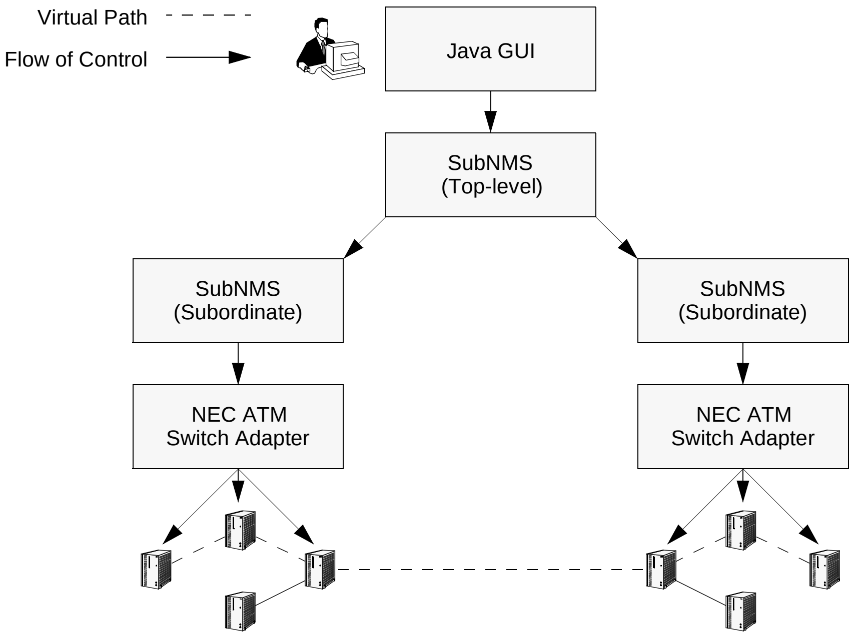

Figure 6: Setting up a VP Using the Prototype NMS

To test the full functionality of the deployed system, a request was made through the top-level SubNMS to establish a virtual path (VP) between two access points, where each access point was associated with an ATM network element from both sub-networks being managed.

The request was directed to the north-facing CORBA server interface of the top-level SubNMS. In the CORBA server component, it was translated into a request onto the domain-independent C++ interface offered by the NMS Core component. The routing function was then activated: this determined that a route could be established across the two sub-networks that represent the managed network domains controlled by the subordinate SubNMS systems. The request to establish the individual connections across both domains was then dispatched to the subordinate SubNMS systems in parallel. The thread performing the path setup then blocked waiting for both requests to complete.

The request was received by each subordinate SubNMS simultaneously, and was processed in parallel on both management stations. Each SubNMS activated its own routing function, and found a route between the supplied access points within their own network domain. This translated into setting up sub-network connections across the sub-networks representing the ATM network elements. Once the routes had been found, each SubNMS then dispatched the requests to establish the required cross-connections to their associated NEC ATM Switch Adapter in parallel and again waited for all the requests to complete.

The NEC ATM Switch Adapters translated the cross-connection requests into a series of M-Creates and M-Actions which were directed to the OSIMIS/ISODE agents controlling each ATM network element. The appropriate configuration changes were made to the ATM hardware—the successful operations were reported back to the Adapters, which in turn passed control back to the subordinate SubNMS systems. They likewise reported success to the top-level SubNMS, which reported to the original caller that the VP had been successfully created.

Once each SubNMS had successfully established the required connections in the underlying domain or Adapters, the network model each maintained was updated to include an appropriate representation of the path or cross-connections that had been created. Details of all existing and newly created paths are visible through either the north-facing OSI agent or CORBA server interfaces.

necessary to communicate with the OSIMIS/ISODE agent. HP's ORBPlus was used for the north-facing CORBA interface to the Adapter.

5 DEPLOYMENT

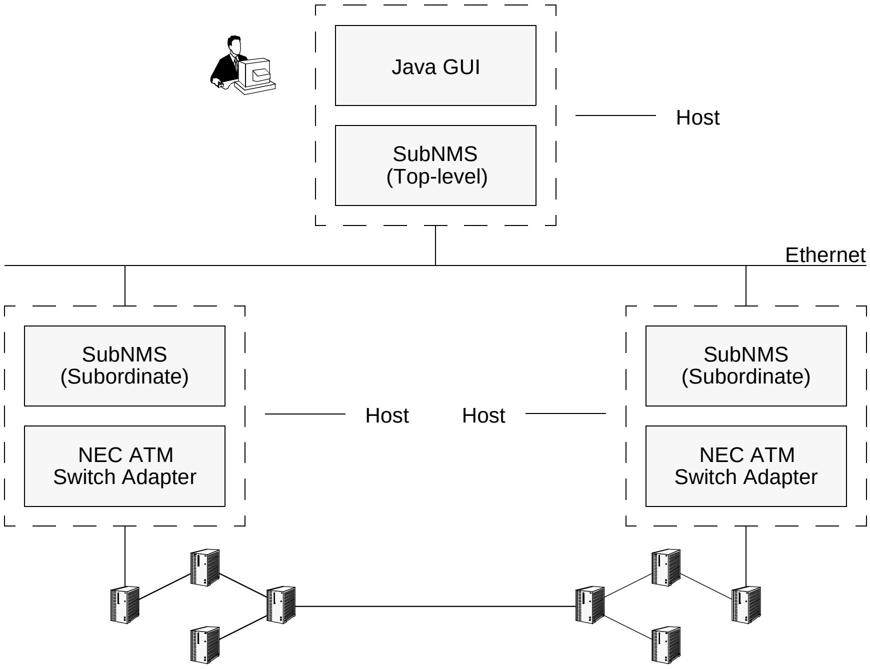

Figure 5: The Prototype NMS Deployed in the Test Network

The deployment used to demonstrate the viability of the NMS architecture consists of a two-level management hierarchy, managing a network of NEC ATM network elements.

The NEC ATM network elements were divided into two four-node sub-networks. Each sub-network was managed by separate SubNMS processes, which were run on separate hosts.

Co-located with each SubNMS was an NEC ATM Switch Adapter, which was used to communicate with the OSIMIS/ISODE agents. There was one such OSIMIS/ISODE agent for each ATM network element, running directly on the ATM hardware. The ATM network elements from each sub-network were connected to their management station via TCP/IP running over an ATM virtual path, which was pre-configured for management communication. The NEC ATM Switch Adapter communicated with the OSIMIS/ISODE agent using RFC1006 (CMIP over TCP/IP).

Each ATM sub-network could be managed directly by interacting with the SubNMS responsible for that sub-network, through either its OSI agent or CORBA server interfaces.

A third SubNMS was then installed on another management station. It was configured to manage the two sub-networks modeled by the existing SubNMS processes. This management station was connected to the other SubNMS stations via TCP/IP running over Ethernet: the top-level SubNMS interacted with the subordinate SubNMS systems through the subordinates' north-facing CORBA server interfaces.

The view presented by this top-level SubNMS, through its own north-facing OSI and CORBA interfaces, was that of two ATM network domains, with no details visible. That is, the individual ATM network elements could not be discerned through this interface. The view exported from the subordinate SubNMS systems to the top-level SubNMS consisted of an ATM "cloud"—only the external, connectable access points were visible.

A simple Java GUI application was implemented to interact with each SubNMS. Iona's OrbixWeb was used to allow the Java GUI to communicate with the SubNMS systems' north-facing CORBA server interfaces, using IIOP running over TCP/IP. The Java GUI could be associated with the top-level SubNMS, to allow the high-level domain view—or it could be associated with the subordinate SubNMS systems, to see their more detailed element management view.

component is implemented. This presents the generic, north-facing interface, which allows it to be integrated with the NMS Core—it implements this interface by communicating using whatever protocol is appropriate for the entity being managed (this could be anything, including TL1, SNMP, CMIP or CORBA). The NMS can thus be extended to support new equipment or underlying management systems by implementing new Adapters, and associating them with the NMS Core.

4 IMPLEMENTATION

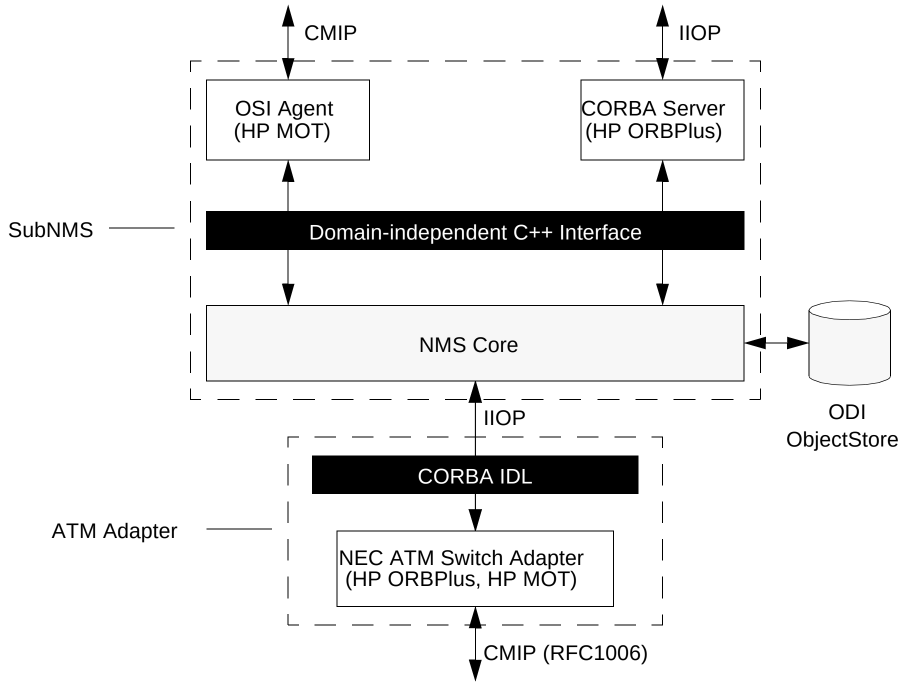

Figure 4: The NMS Implementation

A realisation of this architecture has been implemented in a prototype. This prototype consists of the following software components, each of which has been implemented in C++:

- **SubNMS**:
  The SubNMS incorporates the NMS Core, OSI agent and CORBA server components. It runs as a single UNIX process. Our implementation was initially done under Solaris 2.5, and has since been ported to HP-UX 10.
  The domain-independent interface to the NMS Core was modeled using C++ objects. Both the OSI agent and CORBA server components integrate with the NMS Core through this interface. The network model maintained by the NMS Core is persistified using ODI's ObjectStore ODBMS.
  The OSI agent component is implemented using HP's MOT (Managed Object Toolkit). This tool allows for the rapid development of OSI agents, and proved to be very effective in constructing this component. The OSI agent runs in a single thread within the SubNMS process.
  The CORBA server component uses HP's ORBPlus, a CORBA2-compliant ORB implementation. The facilities offered by this ORB—especially the object activation mechanism, allowing for the on-demand instantiation of object implementations—allowed this component to be developed quickly and effectively.
- **Adapters**:
  Each of the minimal set of Adapter interfaces was designed using CORBA IDL—each Adapter was implemented as a stand-alone CORBA server. All communication with the SubNMS occurs via the ORB.
  For the prototype, an Adapter was constructed for the management of an NEC ATM switch. An OSI agent, implemented using the OSIMIS/ISODE CMIP stack, was provided by NEC to manage the ATM switch. The Adapter was implemented using HP's MOT to generate the OSI manager-side code

connections (a cross-connection across a given sub-network within a layer network). Changes to the contents of the instantiated MIB, as well as fault events (such as communication alarms) are emitted as M-Event-Reports.

No suitable IDL interface has yet been proposed for the CORBA domain. It was therefore necessary to design a proprietary interface, which was based upon our modified version of the I-ETS 300 653 MIB. This mapping loosely applied the JIDM technique of translating GDMO into IDL, however, given that the resulting IDL would be too fine-grained if JIDM was followed exactly, some liberties were taken to make the final IDL interface more useable.

3 ARCHITECTURE

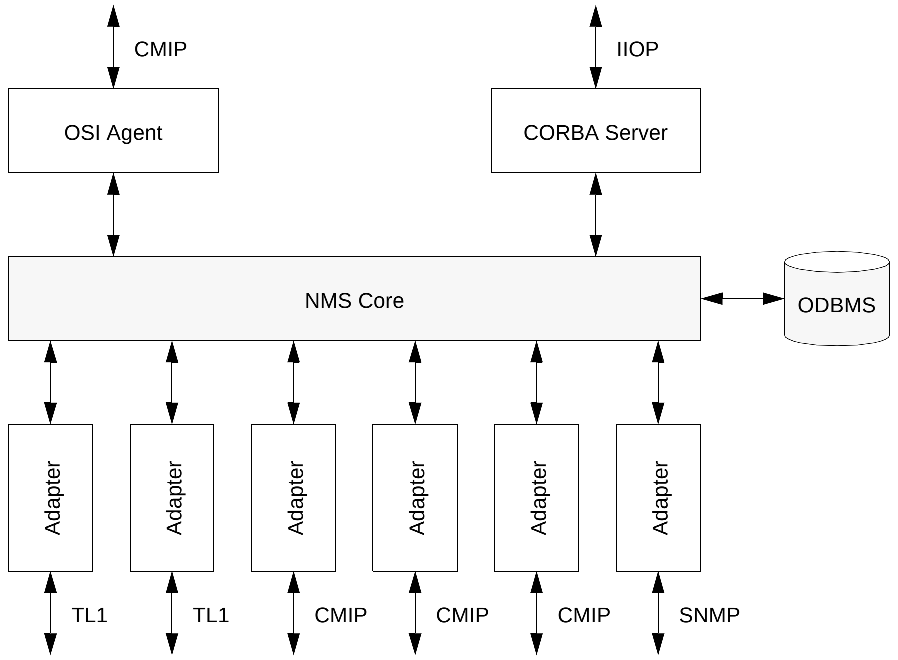

Figure 3: The NMS Architecture

The proposed architecture for the NMS consists of the following components:

- **NMS Core**—This component maintains the generic network model. It offers a domain-independent interface to the OSI agent and CORBA server components. The contents of the network model are made persistent using an object-oriented database. The NMS Core also contains the functionality necessary to perform path management, specifically a routing mechanism for establishing a connection across multiple domains. The core remains abstract, not possessing knowledge of specific vendors' equipment or management systems. Instead, it uses the small set of domain-specific interfaces offered by the Adapter components (see below).
- **OSI Agent**—The OSI agent offers the north-facing CMIP interface, based on the specialised I-ETS 300 653 MIB. The containment tree it maintains is derived directly from the domain-independent interface offered by the NMS Core. Any changes made to the containment tree are translated into modifications in the generic network model through the same interface. Events received from the NMS Core are likewise translated into M-Event-Reports.
- **CORBA Server**—The CORBA server offers the north-facing IDL interface derived from I-ETS 300 653. As with the OSI agent component, the CORBA objects maintained by this components are based directly upon the contents of the generic network model in the NMS Core. Operations on the CORBA objects are passed on to the NMS Core—events received are emitted through an OMG COSS Event Service event channel.
- **Adapters**—A minimal, generic set of interfaces are used to represent the various entities that can be managed, which include SDH and ATM equipment, domain-specific management systems, as well as subordinate NMS systems. To communicate with actual instances of these entities, an Adapter

- **end-to-end connection management**—including automatic, cross-domain routing and equipment configuration
- **fault management**—faults reported from managed equipment would be translated into the generic network model and propagated upwards

The following design goals were applied in this research:

1. Recursiveness

   A hierarchical arrangement of management nodes was chosen to maximise the scalability and flexibility of the deployed system.

2. Reusability

   It was determined that the same software component should be present at each level of the hierarchy, rather than differentiating between leaf, branch or root nodes—this would reduce development and maintenance costs by decreasing the number of individual software components.

3. OSI and CORBA Interfaces

   Rather than restricting the north-facing (to the management component) interface of the management node to a single technology, the architecture would be flexible enough to support multiple management domains in parallel (OSI and CORBA interfaces were initially targeted).

2 NETWORK MODEL AND NORTH-FACING INTERFACE

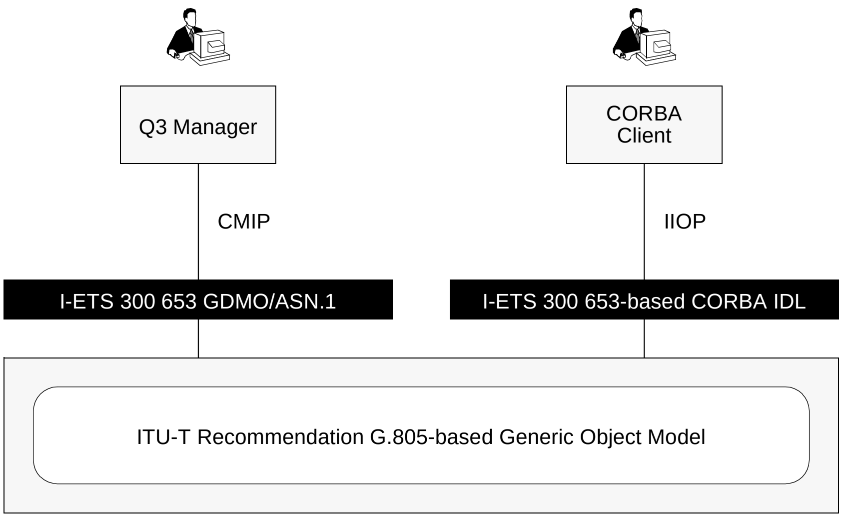

Figure 2: The North-facing Interfaces Provided by the NMS

The NMS would be managing networks from multiple domains (such as ATM, SDH, Frame Relay, ADSL)—it was also desirable to present a single view of the entire managed network. A generic model was chosen to represent all of the network domains, based directly upon the functional architecture described in ITU-T Recommendation G.805.

This model uses a small set of conceptual objects to represent any transmission network. The model is split into layer networks, with each layer representing the capability of transmitting a fixed kind of characteristic information (for example, SDH STM-N signal, SDH VC4 virtual containers, ATM 53 octet cells) across that layer. Each layer contains sub-networks, which may be partitioned into smaller sub-networks. A sub-network represents a collection of connection points, any of which may be cross-connected with any other connection point on that sub-network. A leaf sub-network can represent a piece of transmission equipment (such as an ATM switch), or a subordinate management domain.

For the OSI management domain, a north-facing CMIP interface was provided, using the GDMO/ASN.1 MIB I-ETS 300 653. This MIB provides an implementation of the abstract modeling concepts described in ITU-T Recommendation G.805. While it is suitable for modeling an SDH network without change, some minor modifications were necessary to allow an ATM network to also be modeled. The MIB provides M-Actions for setting up and releasing trails (end-to-end connections across a layer network) and sub-network

Distributed Architecture for Cross-domain Network Management

Jason Etheridge, Graham Chen, Minoru Tanaka[1] and Gen Watanabe[2]

***Abstract*—As the world's communication needs increase, telecommunications providers are building larger and more complex networks to meet this demand. These evolving networks will often contain many technologies, and while most modern networks are still single-vendor, many companies are looking to diversify their networks. While on-the-wire interoperability is clearly defined by various standards bodies, the management of these heterogeneous networks remains mostly undefined.**

Most network management systems deployed today tend to be specific to a single technology domain (ATM, SDH), and are usually limited to the management of a single vendor's equipment. A service provider will use several unrelated and independent management systems to manage the various components of their network—this offers a less than optimal solution.

This paper attempts to address this problem by describing an architecture for a network management system capable of managing such a heterogeneous network, while satisfying the critical requirements of scalability and cost-reduction. A prototype implementation of this architecture is also described, together with an example deployment of the system used to manage a small ATM network.

***Keywords*—Cross-domain network management, I-ETS 300 653, ITU-T G.805, CORBA, OSI, Q3**

***Source of Publication*—Proceedings of the 6th IEEE/IFIP Network Operations and Management Symposium, New Orleans, LA, USA. Feb 1998. ISBN 0-7803-4351-4.**

1 Overview

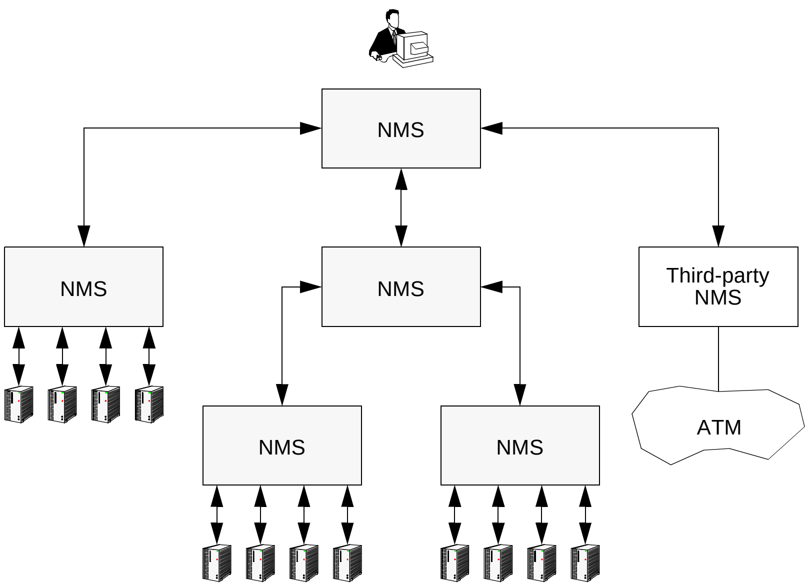

Figure 1: The Proposed NMS to Manage a Heterogeneous Network

The purpose of this research was to formulate the architecture for a network management system (NMS) capable of managing a multi-domain, multi-vendor communication network. That is, the system would be capable of managing a physical network consisting of heterogeneous transmission equipment (ATM, SDH), where that equipment, even within the same domain, could be produced by different vendors.

The functionality of this NMS would include:

- **a network representation**—using a standardised generic model

---

1. Transmission Engineering Division, NEC Corporation, Japan, tan@ted.trd.tmg.nec.co.jp
2. Transmission Engineering Division, NEC Corporation, Japan, gensan@ted.trd.tmg.nec.co.jp