Exhibit 9

A Reliable CORBA-Based Network Management System

Tong Luo, Member IEEE
Tony Confrey
GTE Laboratories, Waltham
K. S. Trivedi, Fellow IEEE
Duke University, Durham

ABSTRACT

Network Management provides the central nervous system for the networks of telecommunications providers. A Telco's Network Management System (NMS) needs to support uninterrupted management functionality of complex networks. The reliability of such systems has direct impact on the quality of services (QoS) provided to the consumers. Even a short down time of the NMS may cause customer dissatisfaction, revenue losses, and may even jeopardize life. In order to expedite the process of transforming technological capabilities into services and to shorten the development cycle of its NMS, the telecommunication industry is adopting CORBA as an underlying architecture. However, neither the CORBA specifications nor the available services currently provides direct support for fault-tolerant objects. Consequently, NMS developers using CORBA must provide their own fault-tolerance mechanism for mission-critical objects. This paper reviews available fault-tolerance approaches in the research literature, presents the architecture of GTE's next generation NMS, discusses the reliability issues involved in such systems, and provides our approaches to solve them. Specifically, we present in detail our fault-tolerance approaches for the naming server, event channels, and other inhouse built critical business objects. A brief comparison of our approaches with others is also given.

I. INTRODUCTION

The distributed nature of telecommunication management systems requires a distributed architecture. There are several distributed object communication standards using OOD technology. These include OMA [1], COM [2], and DSOM [3] [4]. CORBA/OMA, however, has become increasingly ubiquitous in the development of large, cross-platform, distributed systems. CORBA is an open standard and is supported by major software vendors. CORBA simplifies the development of distributed applications by supporting a platform and language independent distributed object execution environment, and by making the communication between distributed objects transparent to the application developers. The componentized architecture supported by CORBA is very attractive to the telecommunication industry because it facilitates the integration of new systems with the legacy systems already in use. The transparent communication between the distributed objects supported by CORBA lets application developers concentrate more on business logic than on system level communication primitives. Thus the development life cycle is shortened, and the development risks are reduced.

In today's increasingly competitive, deregulated and data-centric telecommunication industry, the Telco which survives will be the one that can rapidly transform capabilities available within its own and competitors' networks into bundles of managed service offerings. This requires a higher level of integration and cooperation amongst a Telco's Operational Support Systems (OSS). The network management system (NMS), seeing as a bridge between customer facing systems and the network, is a key component in creating and assuring these services. GTE's current NMS, TONICS [5] [6] [7], provides a unified service level view of large numbers of heterogeneous network devices, and contains a model of the network, its components and their relationships. While an excellent stand-alone management system, TONICS does

0-7803-5284-X/99/$10.00 © 1999 IEEE.

not provide the levels of integration with other enterprise OSS's requested for the future. Therefore, GTE is proceeding to develop a next generation management system. Our next generation NMS will be characterized by its ability to provide the following:

- A componentized architecture, leveraging COTS products and supporting independent design, development, testing, and deployment of components.
- Open, extensible, secure access to network services by OSS's and by customers, competitors and partners.
- Web enabled platform independent client components.

GTE's next generation system is based on Telecommunications Management Network (TMN) [8] [9] layered element and network management components. The Element Management Layer (EML) isolates the upper layer systems from the transport and protocol used to manage the devices and adapts any specific network element model into a common internal model. The EML system makes its services available on a secured CORBA bus. The Network Management Layer (NML) is composed of a set of cooperating CORBA components which perform functions such as service assurance, service provisioning, inventory management, testing and fault isolation, ticketing, etc. The use of an open component based framework supports the use of Commercial Off The Shelf (COTS) products where they are available. A set of federated Java[1] applets provide the user interface for the system. Given the semantics contained within CORBA's Internet Inter Orb Protocol (IIOP) we have developed the ability to provide encrypted, authenticated, authorized and audited access to our CORBA services. This enables us to project both core functionality and user interface displays outside the corporate firewall.

The NMS needs to support GTE's business 7 days a week, 24 hours a day. Reliability and availability of such system is of critical concern. Though CORBA provides a suitable open architecture for distributed applications, neither the CORBA 2.0 standard [1] nor the existing CORBA services [10] provides support for fault-tolerant objects [11] [12] [13]. This is because neither of them specifies the protocols for object replication and recovery, or addresses complex problems such as group communication [14], partial failures [15], and causal ordering of events [16]. This requires that application developers who adopt CORBA as an underlying architecture to build the fault-tolerance mechanism by themselves if high reliability and availability is a system requirement.

Recently, several different approaches have been proposed to build reliable distributed systems with CORBA. A "warm standby" idea is proposed by Sheu, et al [11]. Since it only handles two replications of an object, this approach is not scalable to a larger number of replications. In addition, the protocol for handling failed objects is not transparent to the client object. An "integrated" approach is adopted in Orbix+Isis [17] and Electra [18], which extend and modify the standard Object Request Broker (ORB) with group communication mechanisms. This approach keeps the replication of objects transparent to clients. A drawback of this approach is that it is ORB dependent (implemented with IONA's Orbix). Also it does not comply with CORBA's philosophy that the architecture should be generic and simple, with special requirements being added on as separate services. Yet another approach is the "service" approach [19], which provides the group communicatiou mechanism on top of a standard ORB. This approach keeps the replication of objects transparent to the client. It is ORB independent and follows CORBA's modularity philosophy. The drawback of using this approach is that there is no COTS software product supporting the service available at this time. GTE is reluctant to invest in building its own group communication service which has potentially long development cycle, under the pressure of budget constraints and the pressure of and short project time frames.

After studying the problem of providing high reliability aud availability in CORBA-based systems, we identify three key issues:

- How to make the fault-tolerance protocol of server objects transparent to client objects.
- How to make the replications of the server ob-

[1] Java is a trademark of Sun Microsystems, Inc.

jects consistent with each other.

- How to make the fault-tolerance protocol scalable to multiple object replications.

The objects in GTE's next generation NMS may come from different sources. Some are from vendors, while others are implemented by ourselves. Also a client may have different ways to interact with these objects. The methods of achieving fault-tolerance are usually different for various kind of objects. We focus our discussion on the fault-tolerance mechanisms for the naming server, event channels, and critical business objects.

The remainder of this paper is organized as follows. In Section 2 we give a brief review of the requirements of GTE's next generation NMS, present the system architectnre, and identify the reliability-related issues. In Section 3, we review different solutions currently available in the research literature. In Section 4, we present our solutions and discuss the implementation issues. Finally, in Section 5, we briefly compare our approaches with others and summarize the discussion.

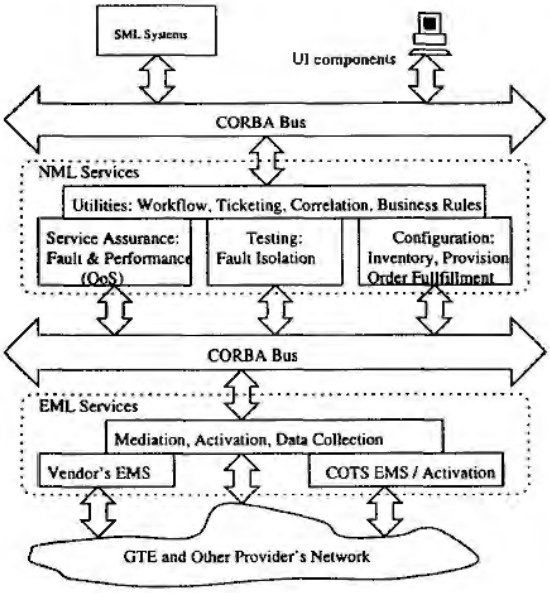

Fig. 1. Logical architecture of GTE's next generation NMS

II. Architecture of GTE's Next Generation NMS

This section describes the architecture of GTE's next generation NMS. The logical architecture, shown in Figure 1, meets many of the needs outlined in Section 1. It provides:

- A set of components each of which can be developed and maintained independently.
- A CORBA based programmatic interface to each of the components, enabling EML and NML services to be accessed by upper layer applications.
- A mechanism whereby most applications can be independent of changes within the network, or at lower layers of the TMN.
- Centrally administered platform independent client components.
- Secure partitioned access to component services via Intranet, Internet, or Extranet.

The architecture is composed of layered components that communicate on a CORBA bus. The lower layers abstract out complexity and provide service to the upper layers. At the lowest layer is the EML system, which provides mediation, data collection and long term storage, connection management, and activation services used by components at the NML.

A. Element Management Layer (EML)

The purpose of the EML in our architecture is to "*be the network*" to higher layer systems, while abstracting out the complexities of interaction with diverse network components. Through a set of CORBA interfaces, it provides a single point of contact which exposes the manageable features of each device in a vendor, transport and protocol independent manner.

For the EML, we are developing the Integrated Element Management System (IEMS) which exposes a network element as a set of CORBA services. These services provide access to the fault, performance, and configuration information and capabilities of the device. The interface is the same whether it is managed using TL1, SNMP, or CMIP; whether management transport is over X.25, TCP/IP, or an OSI stack; and whether the vendor of the device is Vendor A, B, or C. Moreover, since CORBA allows for interface inheritance, insofar as the fanlt behavior of a SONET

device is the same as that of an ATM device, the CORBA interface is the same also.

The IEMS system is composed of four conceptual layers. The lowest one is the network interface layer, which performs mediation and connection management of the physical equipments. This is the module that must encode the details of the management protocol and transport mechanism used to interact with a given element. For some protocols and technologies we use COTS mediation packages, for others we use the vendor provided element manager, for yet others we need to provide hand coded scripts and network interface processes. The NI layer provides service to the adaptation layer mapping or adapting the potentially minimal model presented by the element into our logical internal model. The model layer provides TMN based logical network element and logical event model hierarchies. All upstream components within the EML and above operate on these logical entities. In this fashion, they are shielded from changes within the network and also have a common model for communicating and reasoning about the network. The topmost layer of IEMS is the service layer which provides fault, performance, configuration and other services. These service components operate on logical elements and events, perform long-term information storage and querying, and provide the CORBA services interface to the system.

B. Network Management Layer (NML)

As a user of the EML, the NML forms its requests in the general way so as to minimize its sensitivity to changes within the network or within the EML system itself. The developers of an NML fault management system who need to display device alarms on a graphical display would design the system such that it operated on a generic network element object within the EML interface. This generic interface can provide all the basic alarm information on a given device of any technology type. Thus when new network technologies are added to the network, and supported at the EML, zero changes are required within the upstream system to display alarms from the new technology since the IEMS interface on which it depends does not change.

NML components include:

- Service Assurance, i.e. fault and performance management. This component provides the function of monitoring the network health and proactively detecting potential faults.
- Testing and Fault Isolation. This component, in the event of a network error, helps isolate and diagnose the problem.
- Configuration. This includes inventory management, provisioning and order fulfillment. This component allows us to discover, record and change the state of the network.
- User interface. This is implemented as a set of Java classes, which are clients of the EML or NML CORBA interfaces and provide graphical views which interact with the service assurance, testing and configuration aspects of the network.
- Secure access gateway. This enables secure access into, or between, components. Based on IIOP, this module provides encrypted transport, user authentication, authorization services for access control, and auditing of all gated operations.

For the NML, we are currently developing the NeMoW (Network Management On the Web) system. Given the breadth and depth of functionality required for the NML within a Telco, it is hard to see how a single system developed by a single group could be produced to meet these needs. Within GTE, the NeMoW system is composed of a set of heterogeneous components, each performing a different function, written by a different development group, possibly on differing platforms and languages. What ties these systems together is agreement on a few basic concepts. These systems are component based, each performs its function as independently as possible from the others. They use the services of the IEMS to operate on, and communicate with the network. They share the same set of logical network models, with extensions necessary to the function they perform. Finally they share a common identification scheme, either directly or through the use of a naming service, so that network components can be commonly identified.

The component architecture of GTE's next generation NMS is shown in Figure 2. At the EML, a set

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS
Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS
Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS
Sync your system to PACER to automate legal marketing.