McGill University
Information Networks and Systems Laboratory

An Object-Oriented Approach to Distributed Network Management

Salvatore Torrente

Department of Electrical Engineering,
McGill University, Montreal.

November, 1995

A Thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements of the degree of Master of Engineering

ISBN 0-612-12143-7

Contents

Abstract i

Résumé ii

Acknowledgements iii

Introduction 1

- I-1 Research Objectives and Rational 2
- I-2 Thesis Contributions 3

1 Background Literature 4

- 1.1 Introduction 4
- 1.2 Information Networking 5
- 1.3 Integrated Management Requires Standards 7
- 1.4 Telecommunication Standards 8
- 1.5 Standard Network Interfaces 9
- 1.6 Information Modeling 10
- 1.7 The Object-Oriented Paradigm 11
- 1.8 Objects in Network Management 12
  - 1.8.1 Managed Object Behavior and Notifications 13
  - 1.8.2 Managed Object Structure and Inheritance 14
  - 1.8.3 The Role of Managed Objects in Telecommunications 15
- 1.9 Structure of Management Information 16
- 1.10 Telecommunication Management 17
- 1.11 Intelligent Networks 19
  - 1.11.1 Limitations of IN 21
- 1.12 CMIP/CMIS Communication Elements 22
  - 1.12.1 Generic Network Model 23
  - 1.12.2 Manager/Agent Association 24
- 1.13 Conclusion 26

2 Distributed Network Management 27

2.1 Introduction 27

2.2 Network Management Architecture 28

2.2.1 Distributed Object Database Management Systems 29

2.2.2 Network Management Across Distributed ODBMS 30

2.2.3 Operations Systems and Distributed ODBMS 31

2.2.4 A Distributed Telecommuincation network Model 33

2.3 Prototype Network Information Model 34

2.3.1 Network Element and Equipment 35

2.3.2 Line and Line Termination 37

2.3.3 Channel and Channel Termination 39

2.3.4 Path and Path Group 41

2.3.5 Network and Network Services 44

2.4 Conclusion 46

3 Network Management Prototype 47

3.1 Introduction 47

3.2 Network Management Prototype Setup 48

3.2.1 ITASCA Object Database Management System 49

3.2.2 Prototype Human-Machine Interface 54

3.3 Distributed Management Information 60

3.3.1 Active versus Passive Notification 61

3.3.2 ITASCA ODBMS Scalability 63

3.4 Toward Multiple Management Views 65

3.5 Conclusion 67

4 Future Research Directions 68

4.1 Introduction 68

4.2 Distributed Systems Technology 69

4.3 Distributed Object-Oriented Systems 70

4.4 Information Networking Architecture 71

5 Conclusion 72

5.1 Advantages of a Distributed ODBMS 73

Glossary 75

Bibliography 78

List of Tables

2.1 Management Objectives 32

List of Figures

1.1 Traditional Telecommunication Operations Infrastructure 5
1.2 Relationship of a TMN to a Telecommunications Network 17
1.3 IN Conceptual Model (INCM) 19
1.4 Manager/Agent Association 24

2.1 Distributed Network Management Architecture 28
2.2 Operations Systems 31
2.3 Telecommunication Network Model 33
2.4 Network Element and Equipment Class-Entity Relationship 35
2.5 Network Element Instantiation and Behavior 36
2.6 Line and Line Termination Class-Entity Relationship 37
2.7 Line Instantiation and Behavior 38
2.8 Channel and Channel Termination Class-Entity Relationship 39
2.9 Channel Instantiation and Behavior 40
2.10 Path and Path Group Class-Entity Relationship 41
2.11 Path Group Instantiation and Behavior 42
2.12 Network and Service Class-Entity Relationship 44
2.13 Networks and Services Instantiation 45

3.1 Distributed Network Management Simulation Setup 48
3.2 ITASCA Architecture 50
3.3 Information Model Partitions and Authorization 52
3.4 Prototype Graphical User Interface 55
3.5 Top Level GUI Management Tools 56
3.6 Network Element Creation Tool 57
3.7 Telecommunication Network Service Provisioning 58
3.8 Software Reuse for Interface Development 59
3.9 Performance of ITASCA ODBMS 61
3.10 ITASCA ODBMS Scalability 63
3.11 Separation of Network Element and Network Management Domain . 65
3.12 Distributed Telecommunications Architecture 66

4.1 Telecommunications Information Networking Architecture 71

Abstract

Network management is concerned with monitoring, controlling and co-ordinating network elements for reliable end-to-end customer services. This thesis presents an object-oriented approach to distributed network management of heterogeneous network elements across multiple telecommunication network service provider domains. Specifically, we present a prototype network management approach using distributed object database management systems as a repository and manager of a standard network information model, and we present the results of object interaction across distributed object-oriented databases. In particular, we provide data which illustrates the advantages of an active and dynamic network management environment over static management information bases for fast and efficient telecommunications Operations, Administration, Maintenance and Provisioning (OAM&P) of end-to-end network services. Finally, we expand on distributed object-oriented systems and their role in future information networking architectures.

Résumé

La gestion de réseaux à pour souci le contrôle et la coordination des éléments de réseaux pour assurer aux clients des services de télécommunications bout-à-bout fiables. Cette thèse présente une approche orientée-objets pour la gestion d'éléments de réseaux hétérogènes distribués dans plusieurs domaines de fournisseurs des services de réseaux de télécommunications. Plus précisément, nous présentons un prototype d'une approche de gestion de réseaux qui utilise des systèmes de banques de données orientées-objets employés comme répertoire et gérant d'un model standard d'information de réseaux. Nous présentons également les résultats d'interaction entre objets de banques de données orientées-objets distribuées. En particulier, nous présentons des données qui illustrent les avantages d'un environnement actif et dynamique de gestion de réseaux par rapport aux bases statiques d'information de gérances. Ces avantages assurent une rapidité et une efficacité des operations, de l'administration, de la maintenance et de l'approvisionnement des services bout-à-bout de réseaux de télécommunications. Finalement, nous donnons plus de détails sur les systèmes orientés-objets et leurs rôle dans les architectures de réseaux d'information futures.

Acknowledgements

I wish to express my sincere appreciation and thanks to my thesis supervisor and friend, Dr. Salvatore D. Morgera, for his continued support and constant encouragement throughout this research.

I would also like to acknowledge Wayne Lester, Frank Valente and Bruce Ellacott (Bell Canada, Ottawa, Ont.) for helping me understand the business of telecommunications and network management.

I also wish to thank my colleagues and friends of the Information Networks and Systems Laboratory (INSL) for the wonderful research environment we built together over the years. Special thanks to Albert Pang for his help managing the INSL, and, for his help during the coding and demonstration stages of this research.

Many thanks to my colleagues and friends at Bell Northern Research (BNR) for their encouragement during the final leg of this thesis. Special thanks to Jean-Luc Dupont for his help translating the abstract.

Finally, my sincere gratitude and thanks to my parents and my brother for their constant support and encouragement through the years.

Introduction

The research area of this thesis is the distributed network management of telecommunication resources for efficient Operations, Administration, Maintenance and Provisioning (OAM&P) of end-to-end telecommunication network services across multiple network domains. Traditional network management has focused on simple network monitoring and reporting tasks[1]. Although the monitoring and reporting of problems is important, competing telecommunication network service providers require intelligent systems that not only report problems, but can also take remedial action. Today, an important goal of network management is to support an *integrated approach* to the management of heterogeneous resources and systems. Integration is important because of increasing demand for greater flexibility in network resource management, and, because new telecommunication network services and technologies demand controls not previously required. In today's increasingly competitive telecommunications environment, the goal of integrated management is to augment the management capabilities over new and evolving network technologies, resources and services while reducing the costs associated with operations of enterprise systems. Furthermore, integration across customers, services and networks will be essential for delivering appropriate grades of service to a society moving into the *information age*[2]. As a result, our motivation for research in the area of distributed network management is to demonstrate through analysis and simulations how standards driven *network information modeling* and *distributed processing* can merge in synergy toward an intelligent and integrated management framework for efficient telecommunications OAM&P of end-to-end network services across multiple network domains.

I-1 Research Objectives and Rational

The goal of telecommunications OAM&P is the efficient control and management of telecommunication resources. By *distributed network management* we mean resources monitored, coordinated and controlled by management processes, a desire to share and distribute OAM&P processing and data among management systems, and the need for standard interfaces between multi-vendor network elements[3]. Our research builds on the rational that a standards based *network information model* composed of autonomous and programmable network elements, containing knowledge of the network, is essential for distributed network management of end-to-end network services across multiple network domains. We believe that through autonomous network elements we can *program* intelligence within each element, and, by creating a network composed of many individually programmable network elements, the network not only contains intelligence but can also be *self-healing*[4]. Achieving a balance between network information modeling power and optimal database performance is still, however, an elusive goal. Hence, our research objectives are two-fold:

- build knowledge of a telecommunications network in the form of a standards based network information model within an Object Database Management System (ODBMS) environment;
- measure the responsiveness of information exchange between telecommunications OAM&P *processes* and distributed Object-Oriented Databases (OODBs).

Here, we view each OODB as a *repository* of telecommunications OAM&P management information for a subset of network resources, and, we view the ODBMS environment as a *manager* of the network information model which defines the structure of management information. From the above stated objectives, our research goal is to evaluate the role and capabilities of next generation object-oriented databases as distributed management information bases, and, determine requirements for distributed object-oriented systems in future information networking architectures.

I-2 Thesis Contributions

Our interest in network information modeling and object-oriented databases for telecommunications OAM&P emerged through a research project on **Open Distributed Systems**[1] - *Intelligent Networks Of The Future* and a study of open protocols for telecommunication network management. The goal of managing networks using open protocols is to reduce the operations cost associated with different technologies and to increase system interoperability in a multi-vendor environment. Our research project resulted in a co-authored report on object-oriented technologies[5], a prototype management information model and human-machine interface, and, a co-authored publication on network information modeling[6]. This thesis is an extension of the initial research on *Open Distributed Systems* with contributions in network management architecture and object-oriented database performance analysis.

Specifically, we propose a simple network management architecture for telecommunications OAM&P, designed with the concept of open interfaces between Operations Systems[2] (OSs) and heterogeneous network elements. Our proposed network management architecture builds upon a prototype network information model, implemented within a distributed object database management systems environment. The prototype network information model is an extension of the work presented in [6], and is intended as a standard interface for distributed network management across multiple network domains. The distributed object database management systems environment is intended as an **active** and **dynamic** management information model for telecommunications OAM&P of heterogeneous telecommunication resources, network and services. As such, we will present a performance analysis of distributed database transactions resulting from telecommunications OAM&P processes and events occurring within distributed object-oriented databases. In particular, we highlight the potential benefits of an active and dynamic management information model.

[1]Bell Canada research contract, Ottawa, Ontario; K. Wayne Lester, scientific authority.
[2]We view *Operations Systems* and *Operations Support Systems* (OSSs) to be synonymous.

Chapter 1

Background Literature

1.1 Introduction

Today, telecommunication networks are globally evolving to keep pace with rapid technological, regulatory and market changes[7]. Concurrently, customer services are increasing in magnitude and complexity, and evolving in a distributed nature, creating unprecedented market opportunities for telecommunication network service providers capable of *cooperative* and *distributed* network management. In particular, as competitors in an increasingly deregulated market, telecommunication network service providers need to interact as partners, customers and suppliers in a distributed *resource sharing* environment in order to satisfy market opportunities which are often to large for one competitor alone. As a result, the telecommunications industry is increasingly evolving toward a highly competitive market-creation business driven by innovative services and new technologies; a competition which carries a wide range of benefits to the customer: lower prices, new products and a greater variety of services[8]. And, with public and private networks becoming tightly integrated, there is a critical need for telecommunication network service providers and customers to exchange *administrative* and *operational* data. Hence, telecommunication networks and distributed network management are becoming critical issues for delivering flexible customer services quickly, reliably and cost-effectively within a telecommunications industry moving from a *regulated regional* network business providing voice services to a *global competitive* market place providing *information networking services.*

1.2 Information Networking

Telecommunications is evolving toward *information networking*. By information networking we mean a competitive service offering provided to a customer via a *system* of telecommunication resources distributed across multiple network domains. Here, we view *service* to be the reliable and **transparent** transmission, end-to-end, of information (voice, image and/or data) across heterogeneous resources managed by multiple network service providers. Today, telecommunication network service providers are challenged by legacy systems such as *technology-specific* resources, controlled by *service-specific* operations systems, and, managed through *vendor-specific* interfaces[9], as illustrated in Figure 1.1[1]. These legacy systems are internally focused and limited in their interoperability between platforms executing similar tasks.

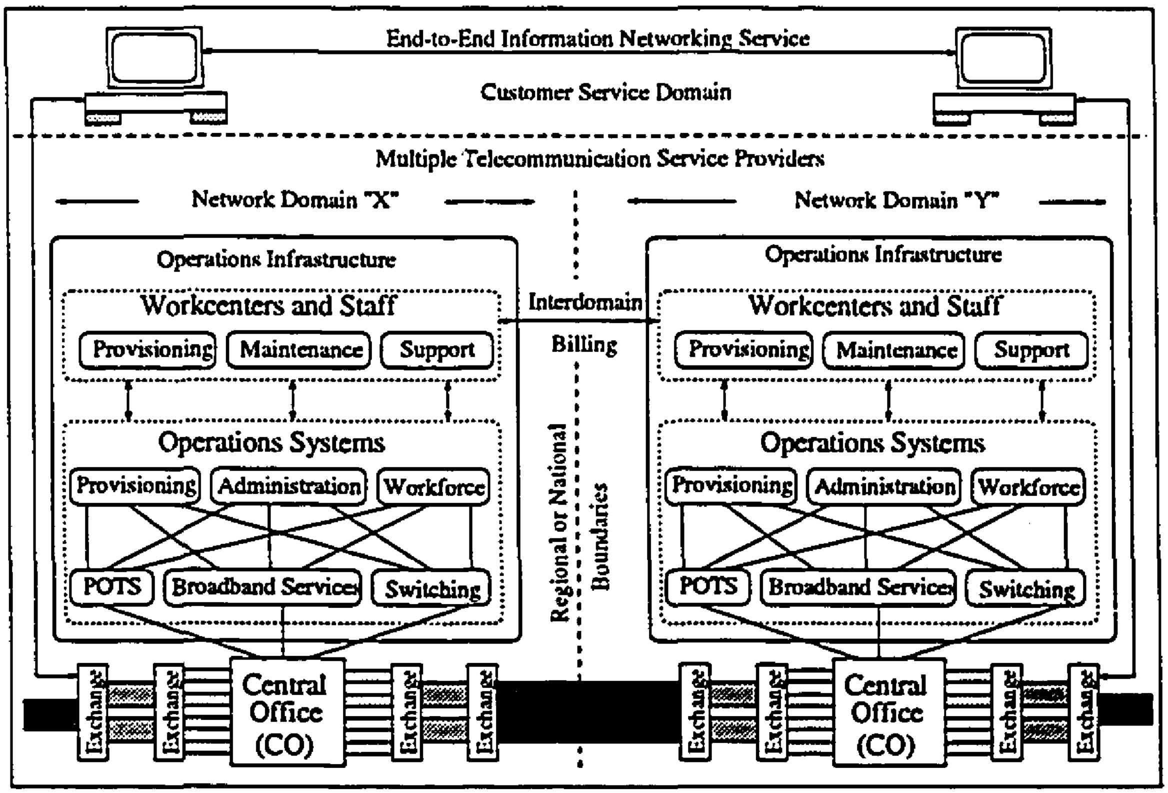

Figure 1.1: Traditional Telecommunication Operations Infrastructure

[1]Figure adapted from *Operational Infrastructure - Common Carrier*[10]

Hence, telecommunication network service providers are now faced with large islands of operations systems which duplicate expensive management tasks and are costly to maintain because of the many platforms, operations system types and development environments. In addition, the great diversity of operations systems make todays *operations infrastructure*[2] an unfriendly environment to human operators because interfaces are inconsistent and data is frequently inaccurate. Furthermore, management of geographically dispersed telecommunication resources is usually accomplished by means of remote login connection, in which case interaction is controlled by the application of the distant resource equipment. In some instances, terminal emulation is supported in which case some form of limited user display is possible; however, the decision making process is usually entirely executed by the human operator. As a result, network management of multi-technology and multi-vendor telecommunication resources is difficult, time consuming and limited, since human operators must be knowledgeable about all types of equipment and interfaces.

Hence, the current state of traditional telecommunication operations infrastructures are a bottleneck for developing and deploying new cost effective services. In order to meet the technical and business challenge for competitive end-to-end information networking services, there is a growing need to separate telecommunication *service design* from *systems design*, i.e., the need to separate the concerns of network technology details from service-related functions and user interface details from core system functions[11]. We believe this separation of concerns is the cornerstone of *integrated management* across heterogeneous telecommunication resources, networks and services. By integrated management, we mean the ability to exchange information via standard and open interfaces for efficient management of end-to-end information networking services[12].

[2]Refers to all the computing, communications facilities, and work centers that are used for operations, administration, maintenance, and provisioning of telecommunication resources[10].

1.3 Integrated Management Requires Standards

In today's global competitive markets a competitor's success will be measured by its responsiveness in acquiring and integrating new technologies within its operations. As resource providers develop proprietary equipment for customer support and resource provisioning, the telecommunications market is seeing an increase in operations systems and network elements from multiple vendors and an increase in unique communication interfaces between these operations systems and network interfaces. The creation of unique and limited interfaces becomes a great barrier for distributed interoperability and systems intergration. To overcome such barriers it is recommended that the telecommunications industry adopt standards[13].

The telecommunications industry stands to benefit from standardization of *interfaces*, designed to achieve interoperability between a broad range of operations systems and network elements. Through standards, resource providers can dedicate more time developing effective network management applications and less time on system interface and management issues. In addition, the proliferation of PC's, LAN's, and similar technologies is decentralizing computer and communication systems to the point where it is impossible to maintain a single-vendor environment; acquisitions, mergers and the combining of diverse networks increases the problem. As a result, the only real solution is a *standards-based* management environment controlling all network and network management aspects, not temporary or expensive work-arounds[14]. Hence, the objective in reaching for a standards-based integrated network management framework is to achieve *flexible control* of new emerging network technologies, network resources and network service interoperability, across heterogeneous network domains – *Open Systems*. And, to a large extent the practicability of open systems is tied directly to the existence of *industry-wide* national and international standards[15]. As such, specific policies must be instituted and implemented to ensure proper telecommunication, interworking and interoperability.

1.4 Telecommunication Standards

Globalization of telecommunications has introduced new industry players and alliances into the world communications market, alliances which are pushing the boundaries of national and international telecommunications policies[16][17]. Success for these alliances greatly depends on their ability to interoperate efficiently across regional and national boundaries. However, the key to achieving interoperability is the availability of compatible national, regional and global communications standards that guide technological change along pathways that satisfy common market interests[18]. Without timely, high quality and harmonized standards at the global, regional and domestic levels, interoperability and efficiency are at serious risk[19].

Standards provide the *keys* that allow components, units, systems and complex networks to interconnect and interwork efficiently and effectively[20]. Standardization transfers proprietary knowledge and technology to the current public domain. Standards also promote mutual understanding and shared knowledge of technology. Defining standards is, however, only a first step, and standards can only make a positive contribution if adopted by a variety of interested parties[21]. In addition, contrary to government regulations, standardization gives manufacturers, service providers and users freedom of choice. Standardization also works to the advantage of resource providers by enabling them to select the most cost-effective equipment from multiple vendors when they are establishing networks. Manufacturers also benefit from standards because standards give them more opportunity to compete for business from all the potential buyers of telecommunications equipment. For these reasons all parties involved in the telecommunications market *should have* a great interest in the standardization process[22]. Today, the main challenge of network management standardization is to develop conventions to support the *integrated management* of heterogeneous networks, resources and systems. Such integration requires that we begin with *standard interfaces* between management systems and the network itself.

1.5 Standard Network Interfaces

Telecommunication networks are managed through *system interfaces*; these interfaces are concerned with the interoperability between physical and logical computing elements and the network environment, they are an integral part of an entire systems design. A network consists of many types of analogue and digital equipment which compose the network resources. Network resources can be customer or provider owned; provider owned resources may include equipment assigned for exclusive use by specific customers. Resources may be physical or logical in nature. Physical resources may include customer or provider systems, their associated subsystems and also the links that interconnect these systems[23]. When managed, the above resources are generally referred to as Network Elements (NEs). Logical resources include communication protocols, application programs, logs, and network services.

Today, existing network elements often include proprietary interfaces which are incompatible with other elements in the network. By interface, we mean the *protocols* defined to exchange management information and *messages* required to monitor, coordinate and control the telecommunication resources. In order for network elements to interoperate efficiently they must interact through well defined and standard interfaces. Standard interfaces provide a common semantic framework of management information which allows telecommunication network service providers to integrate new resources and systems within their network quickly and seamlessly. By common semantic framework, we mean that management systems and network elements share a common knowledge of the protocol in place to exchange management information and a common understanding of the messages available to operate on the network elements, and the messages emitted by these network elements. Furthermore, by implementing standard interfaces independent of the network element design, resource vendors can evolve their network element equipment architecture and computing platforms with minimal impact on systems interoperability.

1.6 Information Modeling

To understand each other, management systems and telecommunication network elements need to share a common semantic framework of management information[24]. By *management information* we refer to the management data and associated operations needed for distributed network management of heterogeneous network elements. *Information modeling* is the basis for achieving such a semantic framework by defining standard interfaces between operations systems and network elements. As applied to systems and network management, information essentially plays two roles[25]: Information associated with management processes and information representing physical and logical network resources subject to management.

Information representing the physical and logical resources of the network, is concerned with such entities as the *abstraction* of communications equipment and systems. Representing this information requires that we model those aspects of the network elements that are of interest and importance for telecommunications resource management. Once this abstraction is achieved, *the totality* of the resources, from the management view point, is represented by that information model. Information associated with management processes is concerned with such entities as customer records, trouble records, log records, etc. Representing this information involves the modeling of the management process and defining the information associated with that management process. Due to the volume, complexity and distribution of the information that needs to be managed, a highly structured modeling method is needed to represent this management information. In addition, since many entities in communications and information networks share common characteristics, this modeling method must be capable of exploiting these characteristics in order to *maximize commonality* and *minimize implementation efforts and diversity.* The above considerations have led the telecommunications industry to adopt the object-oriented paradigm for *information modeling* of heterogeneous telecommunication resources.

1.7 The Object-Oriented Paradigm

The object-oriented paradigm provides a structured environment which attempts to bring order to the chaos of heterogeneous systems and multi-vendor configurations by introducing the modular development of systems *specification* and *implementation*. Specifically, the object-oriented paradigm introduces a new design methodology that is *data-centered*, as opposed to conventional designs that are function-centered[26], and results in a data model based upon the *object* model. A data model is a logical representation of data, and, relationships and constraints on the data. The object model allows developers to model real world resources and entities as a collection of objects that encapsulate data and functions associated with that data within a well defined framework of operations. By *encapsulation*, we refer to the principle that objects can only be impacted by means of messages they receive. Encapsulation demands a clear specification of the *expected* and *allowed* messages at the object's interface and ensures that the operational integrity of an object is preserved. Hence, encapsulation allows the object designer and implementor to change and modify the internal structure of an object without impacting the users of the services these objects provide, and, through mechanisms supporting allomorphism the object-oriented paradigm enables the reuse of knowledge one possesses about other objects. By allomorphism, we mean the ability of an object instance that is a member of one class to be managed as an instance of an object of one or several other object classes. As a result, by its object orientation, the object data model breaks down complex information into smaller pieces. Hence, the object-oriented approach examines the network piece by piece instead of viewing it in its overwhelming and complex entirety. For telecommunications, each individual piece (e.g., resource or logical component) is modeled separately and is, therefore, easier to understand. As a result, object orientation simplifies the management of the complex resources encountered in communicating devices – the main limitation faced in network management today.

1.8 Objects in Network Management

For network management, resources are represented by *managed objects* and management systems exchange information in terms of these managed objects[27]. A managed object (MO) is an Open Systems Interconnect (OSI) defined data model which provides a *conceptual view* of a resource being managed or of a resource defined to support certain management functions. It is an abstraction of a resource that represents the resource properties as seen by management. Each managed object has a responsibility that exists for a limited duration, either to convey knowledge about the environment being modeled, or to operate on other objects in a structured way.

For example, a managed object may be defined to model a transmission line within the telecommunications network. This 'line' MO is part of a system which encapsulates state and behavior. The *state* of the line MO encompasses all of the properties of the transmission line, plus the current values of each of these properties[28]. The state of the line MO is reflected in the set of values assigned to the managed object's attributes. *Attributes* (or instance variables) of the line MO are named characteristics of the managed object[29], within the context of transmission networks. For example, the line MO would include *A Termination* and *Z Termination* end-point (i.e., source and destination) attributes, a *channel* attribute and a *direction* attribute. A value (either single or set) is associated with each attribute of a managed object, and, this value is also an object in its own right. For example, our line MO may represent a connection between Montreal and Ottawa. Here, the end-point attributes (i.e, A Termination and Z Termination) values are the city names (i.e, string type); the channel attribute may be a list of channel MO derived from the line capacity; and, the direction attribute may be an integer from a set (e.g., uni-directional = 1 or bi-directional = 2). Managed object attribute values may be read, set or modified. The rules by which these attribute values can be modified are dictated by the internal constraints imposed by the managed object's behavior.

1.8.1 Managed Object Behavior and Notifications

A managed object can represent not only data, relationships on that data and constraints, but it can also encapsulate operations within the object's boundary, thus providing a uniform framework for the manipulation of arbitrary *user or system* defined data types. Operations on managed objects are performed by passing messages to the object. The message must contain all the information necessary to determine the nature of the operation, how it is to be performed and the conditions for it to be performed. For example, to enable our line MO as a bi-directional connection, a 'SET(*direction*, **bi-directional**)' message would be sent to the managed object.

Since manged objects represent active resources, internal and external events occur that impact the resource modeled by the object and must result in a corresponding behavior in the managed object. Many of these events are of interest to network management. Such events are made visible to the management process as notifications that are emitted by the managed object whenever such an event occurs. For example, the physical medium associated with our line MO may be damaged due to excavation work. The telecommunication resources terminating the line will detect the signal loss and trigger an alarm within the line MO. Since the event requires immediate attention, the line MO must immediately notify the management system of the alarm severity. In addition to the events that occur as autonomous activities of the managed object and its environment, the object also may be impacted by management operations. These events are of interest to resource users and managers, and also give rise to notifications. For example, a network operator may want to monitor the bit error rate of the line to trace a faulty connection. The network operator will instantiate a managed object to collect line statistics and set a threshold at which the managed object will notify the operator. Hence, the behavior defined for a managed object specifies the *dynamic* characteristics of the managed object and its attributes, notifications[30] and actions[31][32][33].

1.8.2 Managed Object Structure and Inheritance

The structure of a managed object is defined by the attributes visible at the MO boundary, the management operations which may be applied to the MO, the behavior exhibited by the MO in response to management operations or in reaction to other types of stimuli, and the notifications emitted by the MO. In addition, managed objects which exhibit similar characteristics are grouped within a common managed object class. A managed object class is a *template* which defines the managed object's characteristics. For example, the line MO may have properties applicable to a wide range of transmission lines. These properties are captured in a 'Line' managed object class, and, a line MO is instantiated for each physical line modeled. Since many instances of a particular managed object class may exist and each instance must be accessible and uniquely indistinguishable, names must be assigned to all objects[3].

Managed objects can be defined by a combination of other managed object classes through inheritance. *Inheritance* is a relationship between classes that permits a new class to be defined by its differences with other existing classes. Specifically, inheritance provides a reusability function where instead of defining all of the characteristics of a new class, the class is said to inherit from another class (called its base class or superclass). Only those characteristics additional to the operations and attributes defined in the superclass need be defined in the new class (called the derived class or subclass). For example, all telecommunication managed objects have a *name* (i.e., unique domain identification) attribute. The name attribute type (i.e., string) can be defined within the 'Top'[4] managed object class which is inherited by all managed object subclasses. Any changes to the *name* attribute type at Top will be inherited by all subclass managed objects. The managed object characteristics that can be derived by inheritance are the managed object attributes (i.e., state), either single or set, and operations and notifications (i.e., behavior).

[3]Referred to as "name-binding" to a Relative Distinguished Name (RDN)

[4]*Top* is a common class name associate with the root node of the inheritance hierarchy.

1.8.3 The Role of Managed Objects in Telecommunications

The purpose of managed objects may seem unclear at first, however, the definition of a managed object precisely identifies the messages (i.e., syntax) used to remotely manage resources and the meaning (i.e., semantics) of the messages[34]. Furthermore, managed objects extend the class concept to include event notification which adds a new dimension to the object-oriented approach of telecommunications network management. For additional flexibility, the OSI data model collects managed object characteristics into modular packages[25]. Packages may be mandatory or conditional. Mandatory packages contain characteristics that are essential and must exist in every managed object that is an instance of a class. Conditional packages reflect those characteristics that are nonessential and may be present in some class members. These characteristics are of interest to management and generally provide some additional capability. These packages are conditional in that the class definition includes a specification of the condition under which these packages must be present. While this modeling approach makes network management systems harder to understand, object orientation and abstraction allow the data model to be *generic* and applicable to all types of resources within an integrated network environment.

Finally, the OSI managed object model seeks to provide a comprehensive framework for handling management of arbitrary complex systems; in addition, it seeks to maximize the information modeling power to handle such complex systems[35]. Despite some criticisms, the OSI managed object model and OSI-based network management is still the only *globally* accepted solution for multivendor network management. As a global solution, OSI provides a *generic standards package* that can be adopted by all vendors, creating a common denominator for connectivity and network management. Hence, these standards provide both public and private networks, and service provider operations infrastructures, with an identical toolset supporting management information exchange between operations systems and network elements[36].

1.9 Structure of Management Information

Managed objects are the means for specifying *standard* interfaces between network resources and a managing system, and, as such, they are essential building blocks for defining the Structure of Management Information (SMI) between open distributed systems. The SMI model plays a central role in the OSI network management approach[35]. OSI network management defines SMI via three core standards documents: (1) Management Information Model (MIM), (2) Definition of Management Information (DMI), and (3) Guidelines for Definition of Managed Objects (GDMO).

The MIM defines how management data must be structured, including managed objects and attributes; what operations can be executed; and, what notifications are required[37]. The DMI defines management support objects and generic attributes which are required by standards[38]. Finally, the GDMO instructs implementors as to what a specific managed object should look like[39]. The GDMO's goal is to ensure consistency among various implementations, which is critical for interoperability and to describe the relationship of object definitions to management protocols. The GDMO uses a language called Abstract Syntax Notation 1 (ASN.1) to define data syntax of attributes and notifications[40]. ASN.1 is a specification language which provides an orderly and clear method for assigning meaning to a collection of data values. It describes data types without specifying a particular data representation or coding rule. In addition, ASN.1 provides a set of Basic Encoding Rules (BER) which specify an orderly data communications transfer and clear data descriptions[41]. Hence, ASN.1 describes syntax for data types and BER describes the actual data values. The GDMO introduces substantial extensions to ASN.1 to handle the syntax of managed information definitions, and, a new language structure called *template* is introduced to combine these definitions. As a result, OSI offers a method to describe data independently of its product implementation.

1.10 Telecommunication Management

The concepts of information modeling and SMI are fundamental to developing a uniform telecommunications management network standard. The objective of a Telecommunications Management Network (TMN) standard is to provide a framework for telecommunication management. By TMN, we refer to a conceptually separate network that interfaces a telecommunications network at several points to send/receive information to/from it and control its operations, as illustrated in Figure 1.2[5]. The concept behind TMN is to provide an organized architecture to achieve interconnection between various types of operations systems and/or telecommunication resources for the exchange of management information using an agreed upon architecture with standard interfaces, including protocols and messages[43].

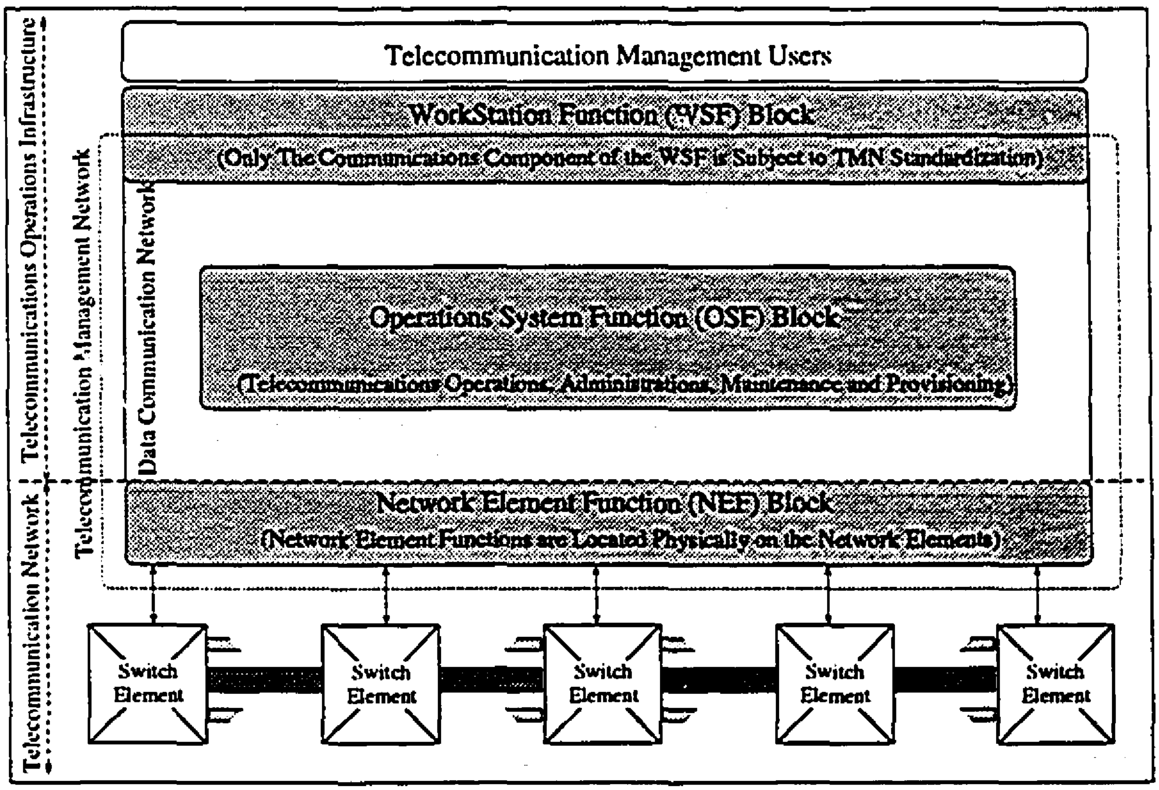

Figure 1.2: Relationship of a TMN to a Telecommunications Network

[5]Figure Adapted from *Principles for a Telecommunications Management Network*[42].

The purpose of TMN is the definition of standard management interfaces. A detailed thirteen step methodology was defined for such interface specification[43]. The TMN standard, as defined by the International Telecommunications Union (ITU[6]), is a generic *management-oriented* architecture intended for management services; TMN, however, does not really define a true *service-creation* mechanism. Within TMN, *management services* refers to a set of capabilities that allow for the exchange and processing of management information, where each management service is built on a set of TMN management service components (i.e., alarm surveillance, network performance monitoring, network status monitoring, traffic control)[44].

TMN has identified some overall rules for defining TMN management services; these services are described by telecommunication management function blocks[42]: The Network Element Function (NEF) block communicates with TMN for the purpose of being monitored and controlled. The Operations Systems Function (OSF) block stores, retrieves and processes information related to management for the purpose of monitoring, coordinating and controlling telecommunication functions, including management functions. The WorkStation Function (WSF) block provides a means to interpret TMN information for management information users[45]. In addition to management functions, the TMN must offer communications between operations systems, and, between operations systems and the various parts of the telecommunications network for management. Here, the Data Communication Network (DCN) provides the ability to exchange management information across the boundary between the management environment and the network resource environment. The principle of keeping TMN logically distinct from the networks and services being managed introduces the prospect of distributing functionality from *centralized* to *decentralized* network management implementations.

[6] ITU-Telecommunications Standardization Sector (ITU-T); previously known as CCITT.

1.11 Intelligent Networks

Although TMN is a generic architecture from a purely management perspective, the Intelligent Network (IN) is a generic architecture intended for various *real-time* connection and management services. The IN is specifically conceived to facilitate and accelerate the introduction of new advanced telecommunication services in a cost effective way. The IN is based on the principle of separating the Network Element Layer (NEL), responsible for information transport, from the Intelligent Layer (IL), containing service execution logic and associated data. The role of the IN architecture is two-fold: first, to centralize intelligent functions in a limited number of nodes, and, second, to sustain a stable and uniform development platform for customer service creation, independent of specific network technology. The IN architecture builds upon the Intelligent Network Conceptual Model (INCM), as illustrated in Figure 1.3[7].

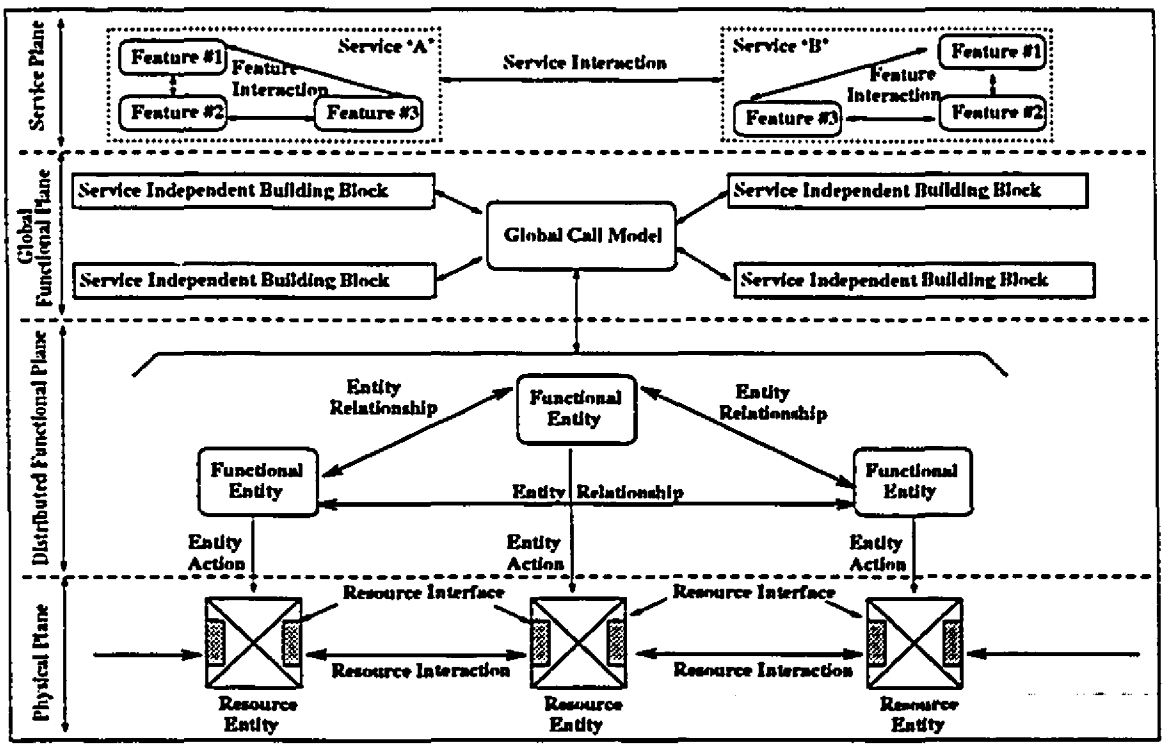

Figure 1.3: IN Conceptual Model (INCM)

[7]Figure adapted from *Intelligent Network Overview - IN Conceptual Model*[3].

The INCM identifies different areas of concerr, represented by a plane; each plane is a particular embodiment of the IN architecture with respect to a particular perspective[3]: The service plane is of primary interest to the service users and providers. The service plane describes services and service features from a user perspective independent of the underlying network. The global functional plane is of primary interest to service designers. The global functional plane describes units of service functionality, referred to as Service Independent Building blocks (SIBs), independent of how the functionality is distributed in the network. The distributed functional plane is of primary interest to network designers and network providers. The distributed functional plane describes the functional architecture of the IN in terms of *functional entities* and *relationships* and both are described independently of how the functionality is implemented or distributed in the network. The physical plane is of primary interest to network operators and telecommunications equipment providers. The physical plane describes the possible physical architectures for an IN in terms of available physical resources and interfaces between physical resources.

The International Consultative Committee for Telephone and Telegraph (CCITT) developed the INCM as a framework for the design and description of IN Capability Sets (CSs) and the target IN architecture, for a particular phase of IN evolution. Each CS is intended to address the requirements of the following areas: service creation, service management, service interaction, network management, service processing, and network interworking. The IN Capability Set-1 (CS-1) is the first standardized stage of IN evolution based on existing technology base and evolution requirements. It defines *capabilities* of direct use to both manufacturers and network operators in support of circuit-switched voice/data services. As a result, although an important goal of the IN is to offer application developers a logical view of call-control and the call model, the IN architecture in its present state of development, is confined to basic telephony call-control capabilities.

1.11.1 Limitations of IN

IN capabilities augment network *intelligence* via a common channel signaling (CCS) network, distributing software control and providing centralized access to shared data[3]. In the conventional IN call-model, however, control of connection types other than point-to-point is complicated. With broadband services increasingly demanding connection types other than point-to-point, the conventional call model becomes inadequate. For example, multimedia services require multiple virtual channels assigned to a single user; thus, a connection model is required which provides flexible virtual channel control and supports a wide range of connection types. In addition, the intelligent network architecture has yet to adequately address the distributed nature of evolving service application software and the need for interoperability with distributed network management, service management and customer premise equipment software. In this respect, it generally is considered that an object-oriented approach should be used for long-term IN development[46]. Furthermore, IN does not yet generally support the property of distribution transparency for service software components, where service implementation is decoupled from network dependencies allowing, for example, the binding of service components across heterogeneous networks, freedom of allocation of components to network nodes, and independence of services from network scale and topology[11].

For these reasons and others, the current IN architecture will evolve to become an Information Network Architecture (INA) with focus of attention progressively moving from the physical network to the software system, with the principles of distributed processing increasingly applied to the software architecture[47][4]. Efforts toward the creation of a common set of interfaces and protocols which clearly separate the switching details from the service aspects of telecommunications networks will continue for both IN and TMN, with a focus on a uniform communication structure for interoperability across resources, networks and services.

1.12 CMIP/CMIS Communication Elements

For telecommunications management, a uniform communications structure is obtained by using one, and only one, communications protocol for interaction between operations systems and between operations systems and network elements. In most cases the Common Management Information Protocol (CMIP)[48] is the most appropriate protocol to transport the management information messages between management systems and network elements[8]. While CMIP describes how to exchange basic management information between open distributed systems, Common Management Information Services (CMIS) provides a logical view of the management information to be exchang d[49]. As such, CMIP and CMIS are tightly coupled for management capabilities between open distributed systems; in particular, CMIS introduces primitives that provide managing systems with *attribute oriented* and *whole object-oriented* operations services. Attribute oriented services allow managing systems to change the value of an attribute (e.g., M-SET) and read the value of an attribute (e.g., M-GET). Whole object-oriented operations services impact an object's behavior as a whole, without being targeted or specifically aimed at setting or retrieving attribute values, although attribute value changes may occur as a side effect. Whole object-oriented services allow managing systems to create new managed objects (e.g., M-CREATE) and to remove them from the structure of management information (e.g., M-DELETE). In addition, notification services such as M-ACTION request an object to perform a certain action on the information model of the telecommunication network and M-EVENT-REPORT let the network resource announce the occurrence of an event. From the managed object attribute specification of the MIM the content of the CMIS services are derived. It is therefore clear which messages are available to the managing system to manage the network resource and which messages are available to the network resource to notify the managing system of events that occur.

[8]The *Simple Network Management Protocol* (SNMP) is also very popular due to its minimal resource requirements. In the sequel, we provide a comparison between SNMP and CMIP.

1.12.1 Generic Network Model

The notion of a management information model is one of the basic pillars in the definition of management interfaces for interoperability. A management information model specifies the Generic Network Model (GNM) of a managed system as seen over a particular interface by a particular managing application; it contains all the object classes that can and will be provided by that managed system to the managing application. By generic, we mean network resources represented by managed objects which reflect the common network elements from the many telecommunication resources with properties applicable across different telecommunications technologies and for various management functions. And, by generic network model we mean a management information model defined to support telecommunications OAM&P functions and data communication between operations systems and heterogeneous network elements[9]. Since telecommunication network elements have attributes that allow users to monitor and control the behavior of network resources and attributes that allow users to monitor and control the relationship between network resources, there is a need to interrelate information contained in various managed objects, not just one managed object at a time. For example, suppose certain managed objects represent established network connections and others the available resources, if a new connection is established, the available resources are no longer the same; hence, there is a *kind-of* dependency between managed objects that a generic network model must specify. A model identifying generic network resources and their associated attribute types, events, actions, and behaviors, can provide a foundation for understanding the interelationships between these resources and attributes, and, can promote uniformity in dealing with the various aspects of managing these resources and attributes. Hence, the generic network model is an essential tool for the generation of uniform configuration, fault, accounting, performance and security management standards for telecommunications OAM&P[23][42].

[9]ANSI and ITU are actively defining a GNM for telecommunications OAM&P [23][50].

1.12.2 Manager/Agent Association

Since telecommunication networks are represented by data (i.e., managed objects), management of a telecommunications network can be seen as an *information processing* activity on that data. Since the environment being managed is distributed, network management can be modeled as a distributed information processing activity between management processes. The OSI network management standard has attempted to define the responsibility between management processes; for a given *management association*, a management process will take on one of two possible roles: a *Manager role* or an *Agent role*[51], as illustrated in Figure 1.4. A Manager is part of a management application that issues management operations and receives notifications to monitor and control managed objects. The Agent is part of a network element application which executes management directives and emit notifications.

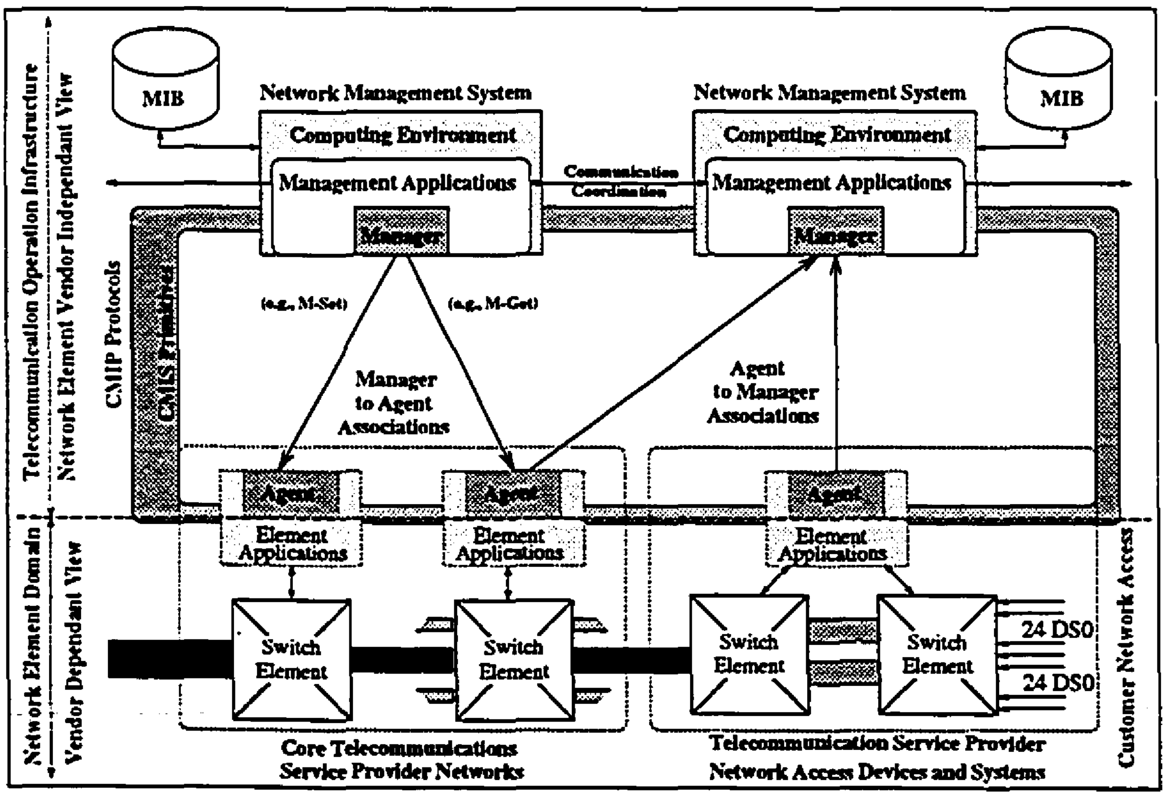

Figure 1.4: Manager/Agent Association

In a telecommunications management environment, a Manager may be involved in an information exchange with several Agents. In this case, issues of synchronization of management directives may exist. Conversely, for telecommunication OAM&P, an Agent may be involved in an information exchange with several Managers. In this case, issues of multiple concurrent directives may exist. An Agent is often implemented by a complex piece of software for a specific network resource technology; however, to use the Agent, this complexity and technology need not be understood, since the Agent may have a well-defined interface understood in terms of a Management Information Base (MIB). By MIB, we mean a conceptual repository of management information defined in terms of managed objects.

This conceptual object (MIB) is actually a database that is shared between *Managers* and *Agents* to provide information about the managed network. For the MIB to be useful, a linkage is provided between the physical reality of network elements and the MIB. This implies that manipulations performed on the MIB will result in real effects on the telecommunication resources. Conversely, information that represents dynamic characteristics of the network resources is to be consistent in the MIB and in the resource it represents; therefore, management information representing resources is manipulated by both the *Manager process* and the telecommunication resource itself. In addition, all management information exchanges between Manager and Agent are expressed in terms of a consistent set of management operations and notifications defined by the MIM (e.g., CMIP Protocols and CMIS Primitives). The managing system and agent system can, therefore, be independently developed, since the MIM defines exactly how the communication is to take place, how the messages are to look and what the messages mean. The MIB concept does not imply any specific form of physical or logical storage for management information, and its implementation is outside the scope of OSI standards[52]. We propose to investigate a physical and logical distribution of management information using distributed ODBMS.

1.13 Conclusion

The next consistent step after TMN is to investigate the incorporation of interoperability interface standards, emerging from open systems environments, into distributed transaction processing, distributed computing and distributed databases so that telecommunication management standards may evolve to support the performance and economic criteria overall network evolution requires. In particular, as new telecommunication technologies and services are introduced, management systems will increasingly need to perform real-time measurements and management of the overall network, and management applications will increasingly need to determine and execute, in real-time, corrective actions based upon events and the current state of the network. As a result, our research program also investigates the incorporation of interoperability interfaces in a distributed ODBMS environment in order to understand and evaluate the performance of distributed transactions in a distributed processing environment.

Our motivations for conducting research into the mechanics and performance of distributed transactions between federated OODBs are two-fold: first, we believe that an OODB is a natural environment for implementing and maintaining an object-oriented network information model for telecommunications OAM&P; second, we feel that a distributed processing environment maximizes the use of human and technology resources; combines customer and provider resources in a cooperative environment; unites networks, computers and operations systems in an indistinguishable way; and helps enterprises downscale complex operations infrastructures into manageable and economically sound environments[10]. As a result, understanding the role, capabilities and limitations of distributed objects in a database management environment is instrumental for distributed network management. Hence, we begin by proposing a network management architecture which illustrates our view of the role played by ODBMS in distributed network management.

Chapter 2

Distributed Network Management

2.1 Introduction

The goal of telecommunications network management is to provide management systems with all the information concerning network resources and to provide operations systems and users with the required operations to configure network connections, maintain and disconnect such connections and bill customers for the service these connections support[53]. To achieve this goal across multiple telecommunication network service provider domains, we propose a network management architecture built upon a prototype network information model for telecommunications OAM&P. The purpose of our prototype network management architecture is to provide users and management systems with a standard interface for efficient end-to-end telecommunications OAM&P through a common network information model. Our prototype network information model is implemented as an *active* and *dynamic* MIM within a distributed ODBMS environment. Within the ODBMS environment, each OODB is modeled as a MIB for a subset of NEs, while the ODBMS is a repository and manager of the standard network information model for telecommunications OAM&P. Furthermore, the prototype network information model is segmented into *private* and *shared* ODBMS partitions for distributed network management across multiple network domains. Here, our research in distributed network management will focus on distributed transactions between telecommunications OAM&P processes within the ODBMS environment.

2.2 Network Management Architecture

Figure 2.1 illustrates our proposed network management architecture for telecommunications OAM&P across multiple network service provider domains. To begin, we view the telecommunications network composed of multi-vendor and multi-technology network elements (e.g., PDH, SDH, ATM). Network elements belonging to a common network technology (e.g. SDH) are represented by a standard *resource* MIM that provides a vendor-independent view of these telecommunication network elements. Furthermore, we view a subset of common technology network elements monitored and controlled by a *network element manager*[43] through a MIB. As opposed to conventional central database approaches, our research proposes *distributed* OODBs as MIBs in a distributed ODBMS environment. Each MIB is a repository of standard resource managed objects and the distributed ODBMS is the software system used to manage, in a transparent way, the distributed MIBs residing at multiple nodes.

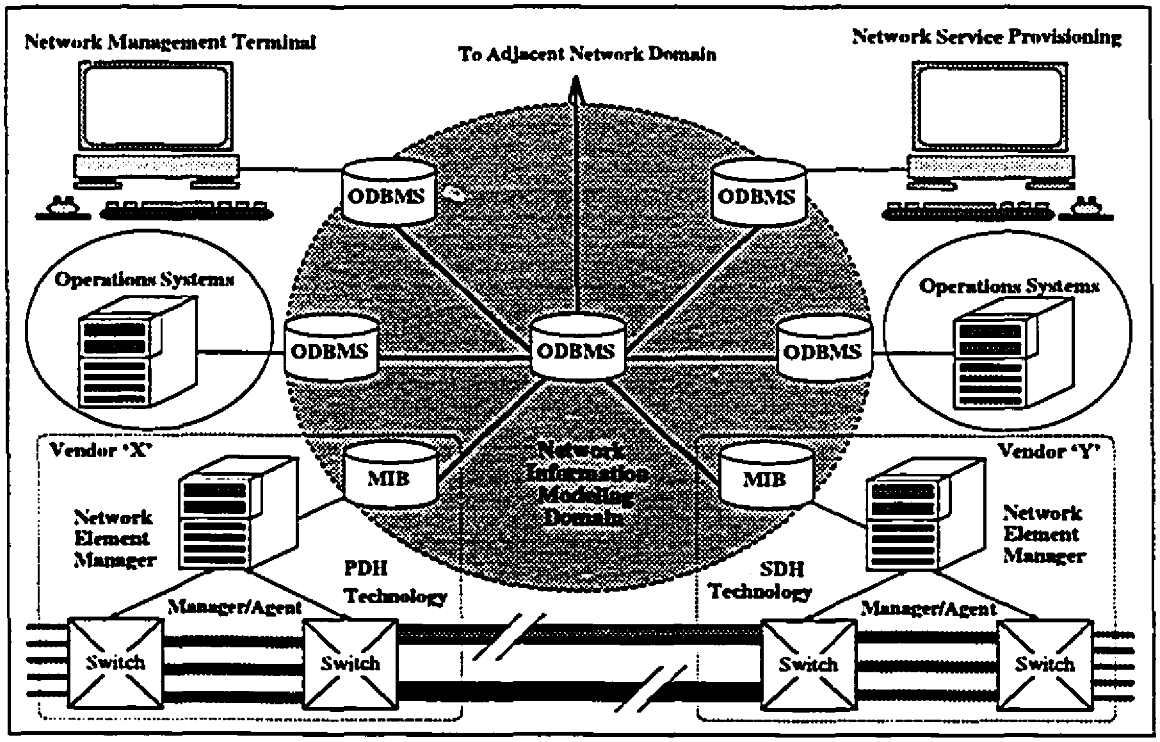

Figure 2.1: Distributed Network Management Architecture

2.2.1 Distributed Object Database Management Systems

OODBs combine the paradigms of object identity, class membership, inheritance and methods from the object model with the database functionality of persistence, concurrency, security and declarative querying thus providing the strengths of both[54]. An ODBMS is a *persistent* and *sharable* repository, and manager, of OODBs for controlled simultaneous multi-user access, and, a *distributed* ODBMS is an OODB environment that allows data stored in multiple physical databases to appear as one centralized database to users and programs[55].

Within a distributed ODBMS environment, the fine granularity of objects, and their ability to form complex objects, leads to a unit of mobility - the object itself. Since objects can be dynamically relocated, the resulting distributed system provides flexibility, and, in particular, enables a more natural modeling of the data based on the structure and location of the operations infrastructure. The potential benefits of distributed ODBMS environments are widely recognized and are described in [56][57][55]: (1) **Autonomy:** Users that share or want to share data can move the data to appropriate locations. (2) **Performance:** Data can be placed close to where it is frequently used, thus reducing communication cost. (3) **Availability:** Access of critical data can be compromised by a crash; since data is distributed, a crash of one site will not make the overall system inaccessible. (4) **Expandability:** Systems can more easily accommodate increasing database size in a distributed system. (5) **Shareability:** The computing environment is distributed, both locally and geographically. (6) **Economy:** It is more cost-effective to build a system of smaller databases with the equivalent power of a single large system. Specifically, each OODB can be physically located on a computing platform configured to meet the specific requirements of users and applications accessing a given OODB. As the user requirements for a specific OODB evolves (e.g., need for greater processing power), only the computing resources on the platform running that OODB need to be upgraded.

2.2.2 Network Management Across Distributed ODBMS

An important objective of the distributed ODBMS environment is to hide users and management systems from the network details. Specifically, since OODBs in a distributed ODBMS environment support the notion of single-image object databases and data transparency, application developers do not need to code differently depending on the object locations. Transparency is an extension of the database concept of data independence, which enables an application to deal with a *logical* data organization independent of its physical implementation. By data independence we mean that databases can be reorganized without breaking applications, and, similarly, by data transparency we mean that objects can be relocated without breaking applications. Data independence and transparency are possible because objects are uniquely identified independent of the object attribute values. As a result, new objects can be relocated without conflict or collisions with existing objects within the ODBMS.

In our proposed network management architecture, *distributed network management* is achieved by passing messages to objects within a distributed ODBMS environment. At the communication level, message passing is performed by a Remote Procedure Call (RPC) style *location independent* object invocation whereby interacting objects reside at different nodes. As opposed to RPC *call-by-value* parameter semantics, however, object references are also passed remotely, leading to *call-by-object-reference* semantics. In addition, single objects or even complete object graphs could be passed as parameters to the callee[57]. By objects, we mean *managed objects* encapsulating telecommunication resource *attributes*. Here, attributes are *triggers* which capture dynamic aspects of the network under the form of event-condition-action rules. When a significant event arises, the associated condition is evaluated and, if the result of the evaluation is positive, a specified action is executed. The resulting ODBMS schema is an active network information model for defining standard interfaces between operations systems and network elements.

2.2.3 Operations Systems and Distributed ODBMS

Figure 2.2 illustrates the relation between *operations systems* and the ODBMS environment for telecommunications OAM&P. To begin, we view operations systems composed of *management systems* interconnected via a distributed computing environment. By *distributed computing environment* we refer to a collection of utilities, languages and libraries which support the development and operation of distributed network management applications. Here, management applications interact with managed objects through OODB interfaces. For example, a network element failure triggers an alarm at the MIB interface. The alarm message may be directed to an application associated with an OODB interface (i.e., alarm reporting application or human-machine interface for diagnostics), or the message may be directed at other managed objects within the ODBMS environment. This opens the opportunity to program intelligence within the ODBMS network information model.

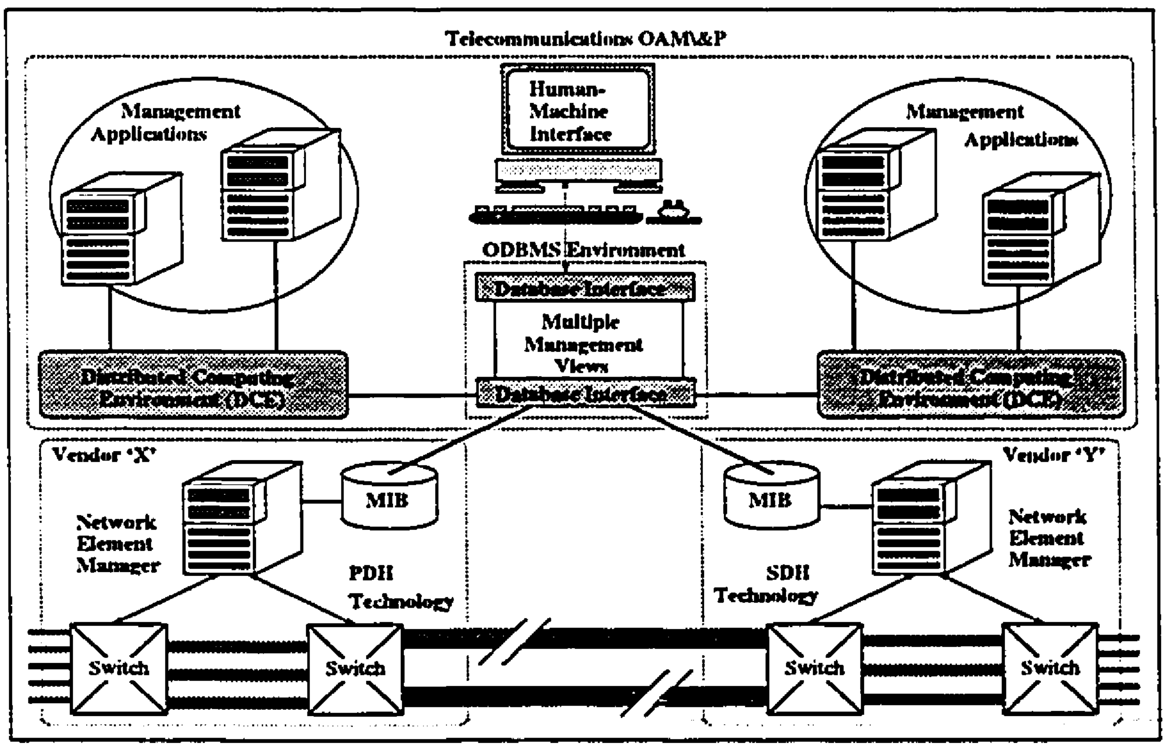

Figure 2.2: Operations Systems

For example, a network designer defining customer networks may program the ODBMS schema to emit notifications to applications residing within the operations systems, and, to managed objects representing network services provided to the customer. In the event of a network element failure, affected service managed objects are marked by the alarm message and a network operator may inspect these objects to evaluate the extent of the failure. In addition, a management application may devise a sequence of alternate paths to route around the failed element and communicate these instructions to the MIB. Since all OODBs share a common database schema, management applications can communicate with any OODB, and the ODBMS carries the message to the managed objects. Since each ODBMS handles only a portion of the overall schema, contention for CPU and I/O services are not as severe as for centralized systems, while expansion is handled by adding processing and storage power. As a result, network management objectives can be automated toward real-time response. For telecommunications OAM&P, management objectives are grouped into five main categories: *Configuration*, *Fault*, *Accounting*, *Performance* and *Security*, as illustrated in Table 2.1. These management objectives are immutable.

Table 2.1: Management Objectives

Category	Objectives
Configuration	To collect/analyze data concerning the state of the managed network in order to design and control the network and alter the operating parameters of the network elements.
	Service: To establish connections for a specified bandwidth, Quality of Service (QoS) and duration. **Resource:** To change the configuration of the managed network for a given service.
Fault	To analyze logs and execute diagnostic tests to detect and isolate problems, and initiate corrective actions.
Security	To provide authorization, access, encryption, authentication and maintenance, and manipulation of security logs.
Performance	To collect/analyze data concerning the state of the managed network in order to evaluate and maintain system performance.
Accounting	To collect resource *usage* data and allocate cost for services provided by these network resources.

2.2.4 A Distributed Telecommuincation network Model

Figure 2.3 illustrates our view of a distributed telecommunication network management model for telecommunications OAM&P. Here, telecommunication network service providers communicate through a common network information model (i.e., standard and public interface) distributed across ODBMS *shared partitions*. The network information model encapsulates a telecommunication *service* MIM representing end-to-end telecommunication network paths and cross connections available at the edge of their respective domains. By interconnecting these shared partitions across domains, each telecommunication network service provider can build an *image* of possible network services and connection functions for end-to-end information networking. The telecommunication service MIM is one management view of an overall network information model; hence, our next objective is to prototype a network information model which defines a common telecommunication service, network and resource management view across multiple network domains.

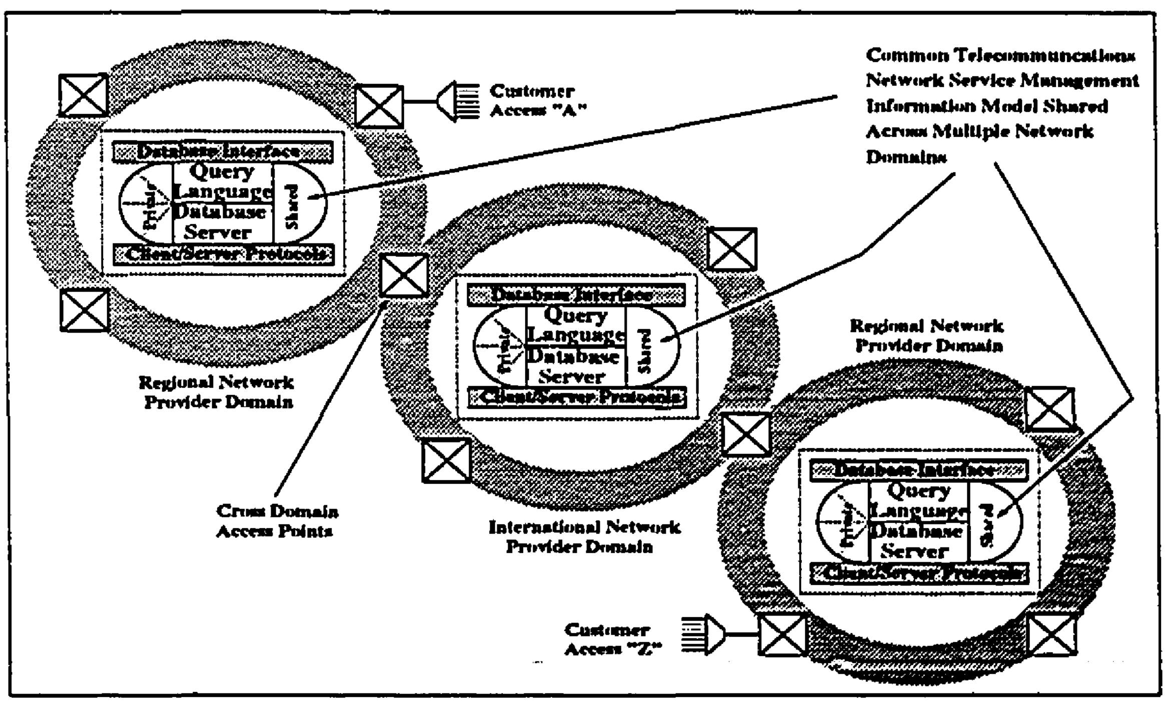

Figure 2.3: Telecommunication Network Model

2.3 Prototype Network Information Model

Our prototype network information model is derived from the ANSI T1.214-1990 Generic Network Model (GNM) defined by the T1M1.5 working group of the accredited standards committee on telecommunications – T1[1]. The T1 committee defines telecommunications OAM&P as a distributed information processing activity that supports the operations, administration, maintenance, and provisioning of telecommunication network services and that is characterized by the cooperation of two or more OAM&P processes located in different real open systems[23]. The ANSI T1.214 is a framework for defining management information exchange; it specifies interface requirements between operations systems and network elements and describes a GNM needed to develop telecommunications OAM&P application message standards. Although the ANSI T1.214-1990 GNM is not the latest release in network modeling, it does contain an initial set of *generic* managed object classes and attribute types defined to support a subset of the fault management, alarm surveillance as applied to digital access networks, digital point-to-point transport media, and common aspects of network elements for modern telecommunication networks.

Our presentation of the prototype network information model will not elaborate on network elements; instead, our work will focus on the *attributes* and *methods* which implement the **active** and **dynamic** nature of the managed object relationships for telecommunications OAM&P. By *relationship* we mean a tuple of managed objects related by some property that pertains to all objects of a tuple. The resulting *entity-relationship* diagram provides a conceptualization of instances of the managed object classes[2]. Although our proposed network information model implements standard attribute *names* from the ANSI T1.214 GNM, their associated values are generated by prototype ODBMS methods.

[1]The T1 committee is a U.S. Regional Standardization Organization (RSO) for the North American telecommunications industry.

[2]Throughout this section class names will be printed in bold and object instances in *italics*.

2.3.1 Network Element and Equipment

Figure 2.4 illustrates the ANSI T1.214 **Network Element** managed object class and its entity-relationship. Network elements represent telecommunications equipment that perform network element functions. For example, a MUX concentrates information from multiple input ports onto high speed lines while a DCS enables cross connections between line terminations for end-to-end services. In our prototype, network elements and lines are connected via circuit packs[3]. A **Circuit Pack** is a managed object class specific to our prototype and is composed of other equipment, such as shelves, cards, ports and data tables. Hence, *Equipment* objects are composite objects built from *Hardware* and *Software* managed objects. A **Hardware** managed object class represents physical components of equipment. A **Software** managed object class represents programs and logical information stored in equipment. Finally, *Equipment*, *Hardware* and *Software* managed object instances may be nested.

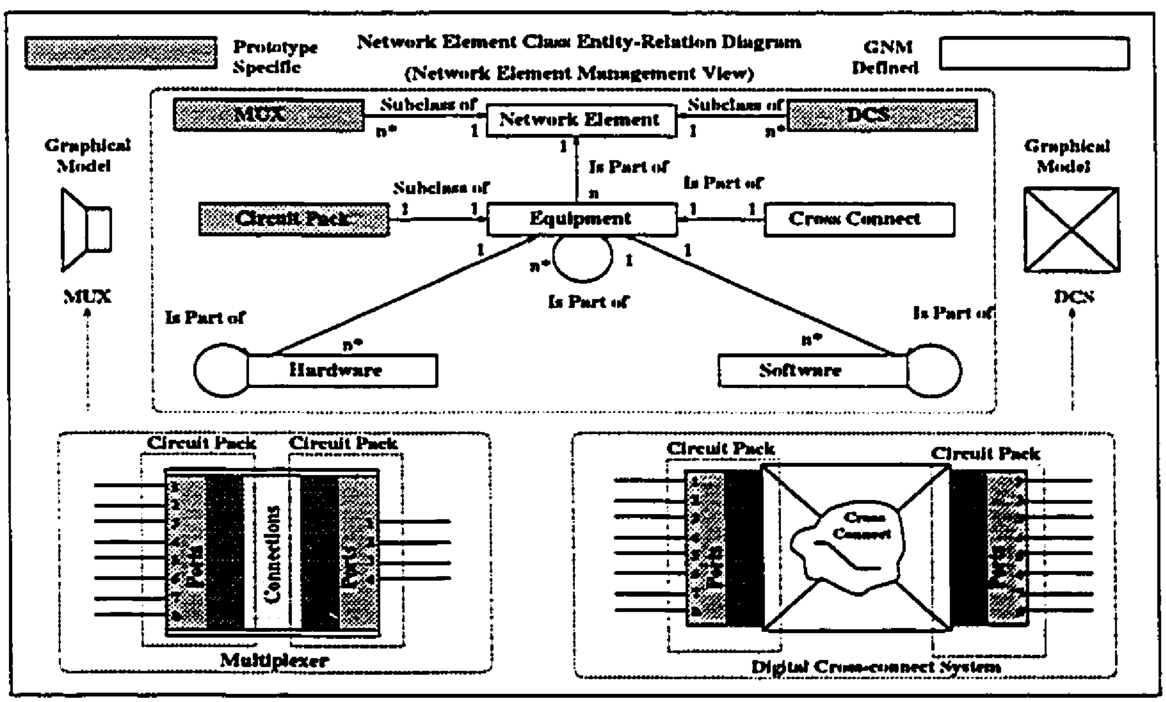

Figure 2.4: Network Element and Equipment Class-Entity Relationship

[3]Arrows indicate the direction and cardinality; n means one or more, n^* means 0 or more.

For telecommunication network elements we defined a Create-NE method which instantiates network element managed objects, as illustrated in Figure 2.5. Here, network element managed objects are characterized by attributes defined by the GNM[4]. Our prototype implements attributes as triggers, where an event (e.g., AlarmState turning ON) will generate a notification within the ODBMS environment and at the database interface. This highlights a crucial characteristic of our network information model. We are looking for an active database schema where managed objects can communicate with other related managed objects independent of the managed object location. For example, the failure of a telecommunication resource will set the trigger of the *Network Element* managed object (i.e., AlarmState is ON). The trigger enables a network element notification method which sets an alarm at the *Network* managed object. Hence, network element alarms and notifications are propagated across the ODBMS environment to the above *Network* and *Service* managed objects.

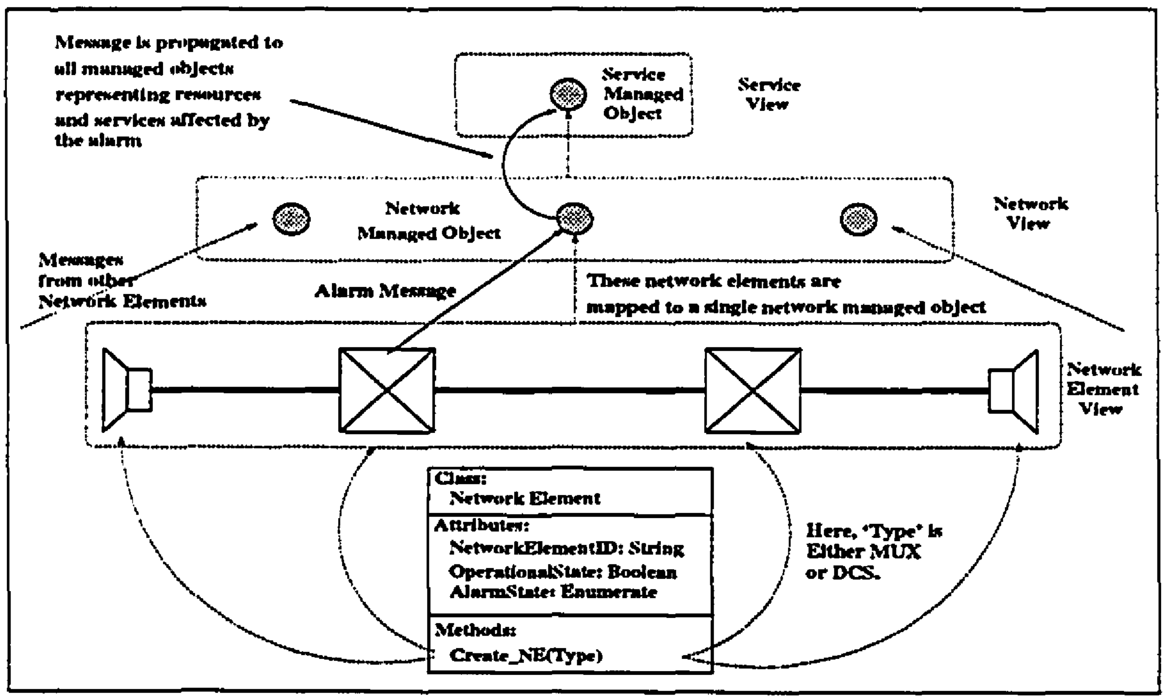

Figure 2.5: Network Element Instantiation and Behavior

[4]This document will focus on key attributes and notifications; see [23] for a detailed list.

2.3.2 Line and Line Termination

Figure 2.6 illustrates the GNM[5] **Line** managed object class and its entity-relationship. A line interconnects network elements through termination points that provide **point-to-point** connectivity between source and destination. A **Line** object class represents a physical transmission medium and associated equipment required to provide the means of transporting information between two consecutive ports. A *Port* is an equipment managed object specific to our prototype. The physical transmission medium consists of an ordered sequence of one or more spans or pairs of spans and intermediate equipment (i.e., repeater or regenerators). And, the *Line Termination* object represents points at which the digital or analog signal is originated, terminated, or both, including the overhead associated with the line. This managed object class is a refined subclass of the parent **Termination Point** object class. A line may be uni-directional or bi-directional and a line operates at a specific rate.

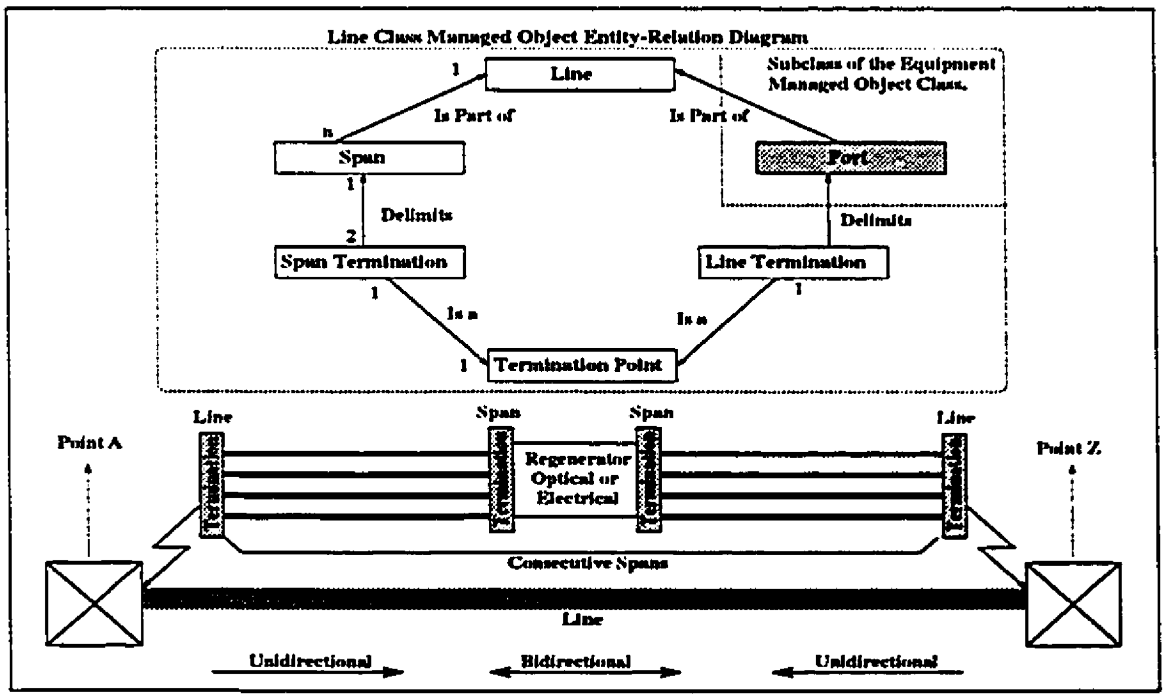

Figure 2.6: Line and Line Termination Class-Entity Relationship

[5]For simplicity we will use the term *GNM* when referring to the ANSI T1.214-1990 GNM.

For telecommunication transmission, we define a Create-Line method which instantiates a *Line* managed object between source (i.e., point A) and destination (i.e., point Z) for a given rate and directionality, as illustrated in Figure 2.7. For our prototype network information model, the source and destination can be either a MUX or DCS managed object. The Create-Line method scans the circuit packs on each network element for available ports that match the line rate; each port becomes a termination point. When a failure occurs at a network element, the end-to-end telecommunication network and service depending on it will be affected; hence, we used the line managed objects and termination points to propagate alarm messages to the adjacent network elements. Each *Network Element* managed object then notifies their associated *Network* managed object and, subsequently, the *Service* managed object. This implies that a failure alarm at a network element can propagate along many different lines cross connected at the network element. These alarms will be directed at the database interface for processing by network management applications.

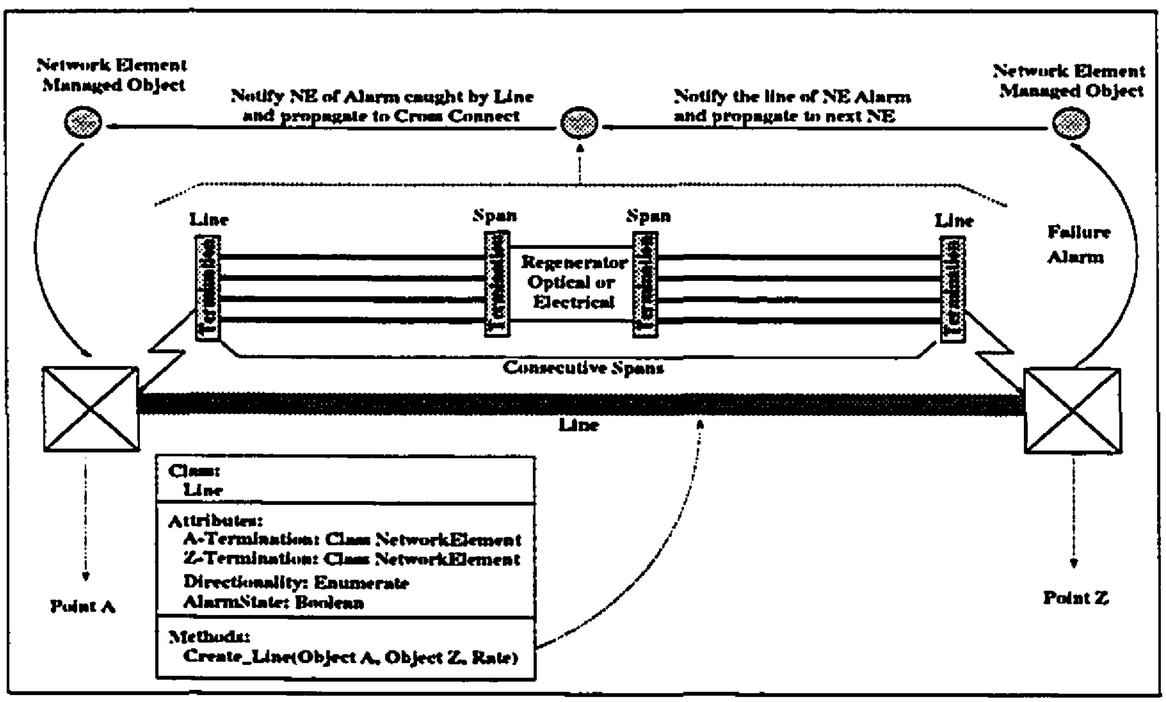

Figure 2.7: Line Instantiation and Behavior

2.3.3 Channel and Channel Termination

Figure 2.8 illustrates the GNM **Channel** managed object class and entity relationship. An instance of this class represents logical objects, called *Channels*, that are connections between two channel terminations, including the channel terminations themselves. A channel termination point is a subclass of the **Termination Point** object class. Channels can be composite objects containing other channels (i.e., subrates) derived from higher rate *Channel* managed objects. A channel may be uni-directional or bi-directional with 'A Termination' and 'Z Termination' end-points. For uni-directional channels, the direction of information transmission is denoted by 'A Termination' and 'Z Termination' attributes. The **Channel Termination** object class is a class of managed objects that represent points identifying portions of a containing path and that delimit a channel. Finally, a channel represents a portion of a path, that is, a sequence of one or more channels linked by cross-connections to comprise an information path or framed path for end-to-end network services.

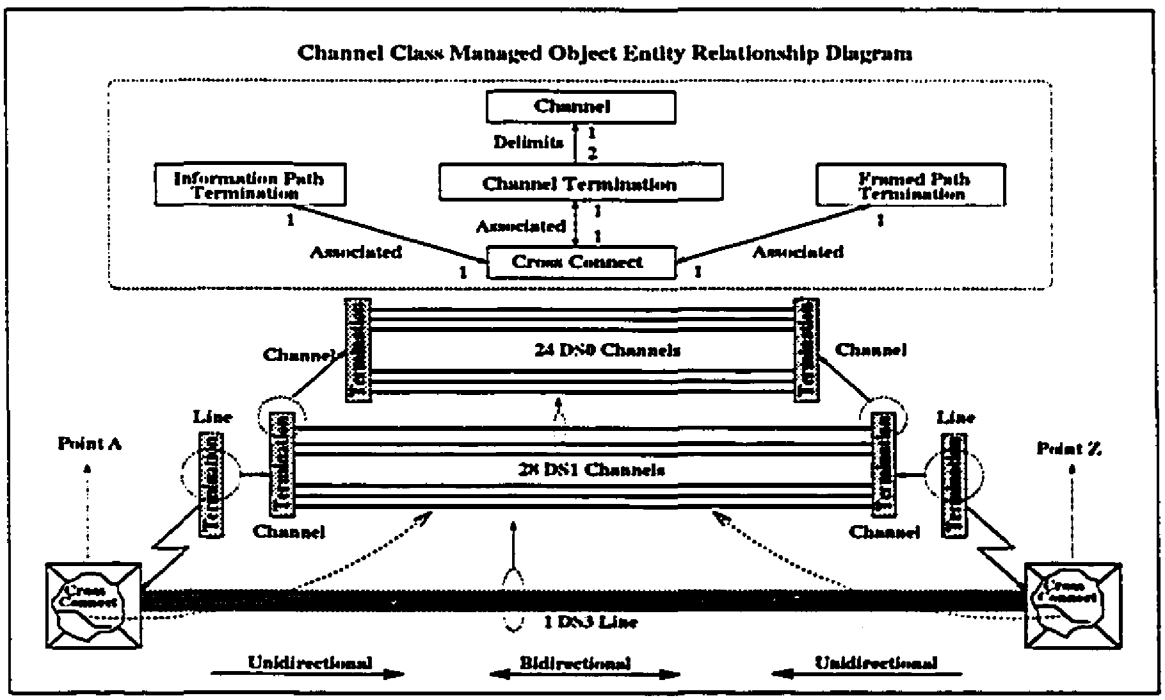

Figure 2.8: Channel and Channel Termination Class-Entity Relationship

Within our prototype network information model, channels are created from a *Line* managed object or a *Channel* managed object. The line class method 'CreateFirstOrderChannels' instantiates a group of channel managed objects associated with a given line. These channels can be referenced as a list of managed objects through a lines 'FirstOrderChannel' attribute. The channel class method 'CreateNextOrderChannels' instantiates a group of channels (i.e., subrates) from a single channel. These channels can be referenced as a list of managed objects through a channel's 'NextLowerOrderChannels', while the parent line or channel is referenced in the 'NextHigherOrderChannel' attribute. In both cases, the sum of all channel rates cannot exceed the throughput of the parent *Line* or *Channel* managed object; for example, a DS1 channel can generate a maximum of 24 DS0's. The 'AlarmState' attribute inherits the notification of the **Line** class. The line and channel 'AlarmState' can be triggered by either a line alarm notification method resulting from a failed loopback test indicating a cable malfunction, or a network element failure.

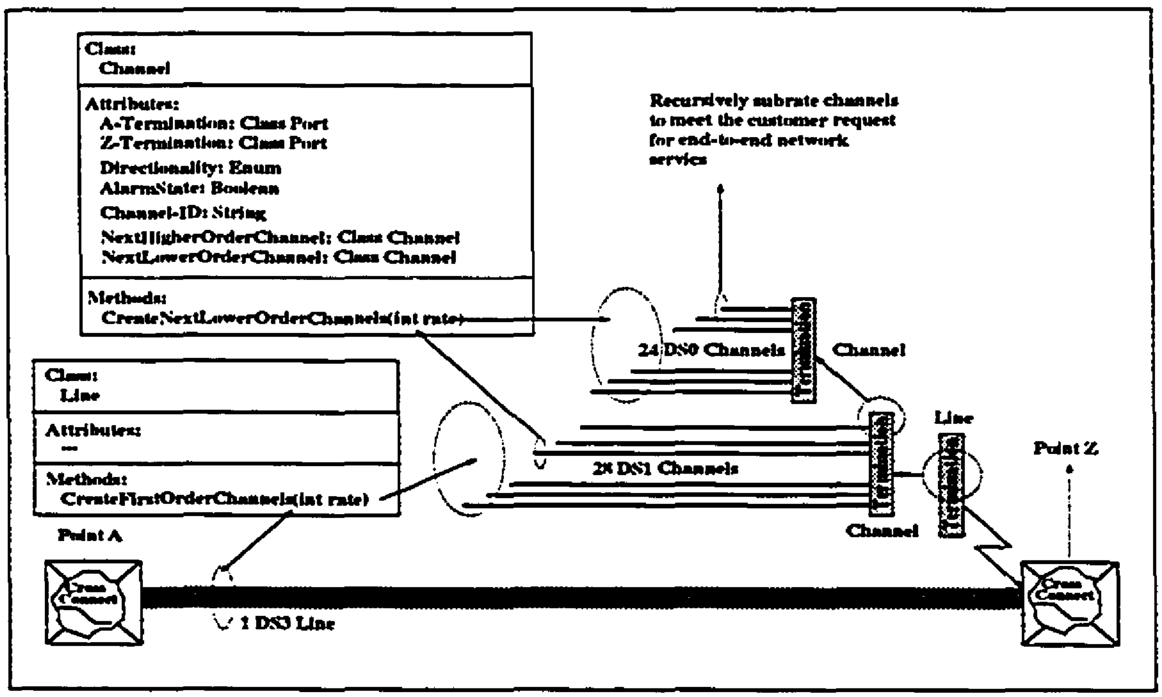

Figure 2.9: Channel Instantiation and Behavior

2.3.4 Path and Path Group

Figure 2.10 illustrates the GNM **Path** managed object class and its entity-relationship. A path represents a logical object that is a connection between two path terminations of the same type, including the path terminations themselves. A path is characterized by a specified rate, determined by an aggregation of channels, independent of the physical means carrying the signal. An information path carries information from point-to-point preserving its content, while a framed path is characterized by a specific rate and frame format. Information paths and framed paths are carried over an ordered sequence of one or more lines, and these lines may be at the same, or higher, rate than the path. A path may be uni-directional or bi-directional between source and destination (i.e., points A and Z). For uni-directional paths, information transmission is denoted by the 'A Termination' and 'Z Termination' attributes. Finally, a **Path Group** object class is a class of managed objects that consists of sets of parallel paths of the same type having common originating and terminating equipment.

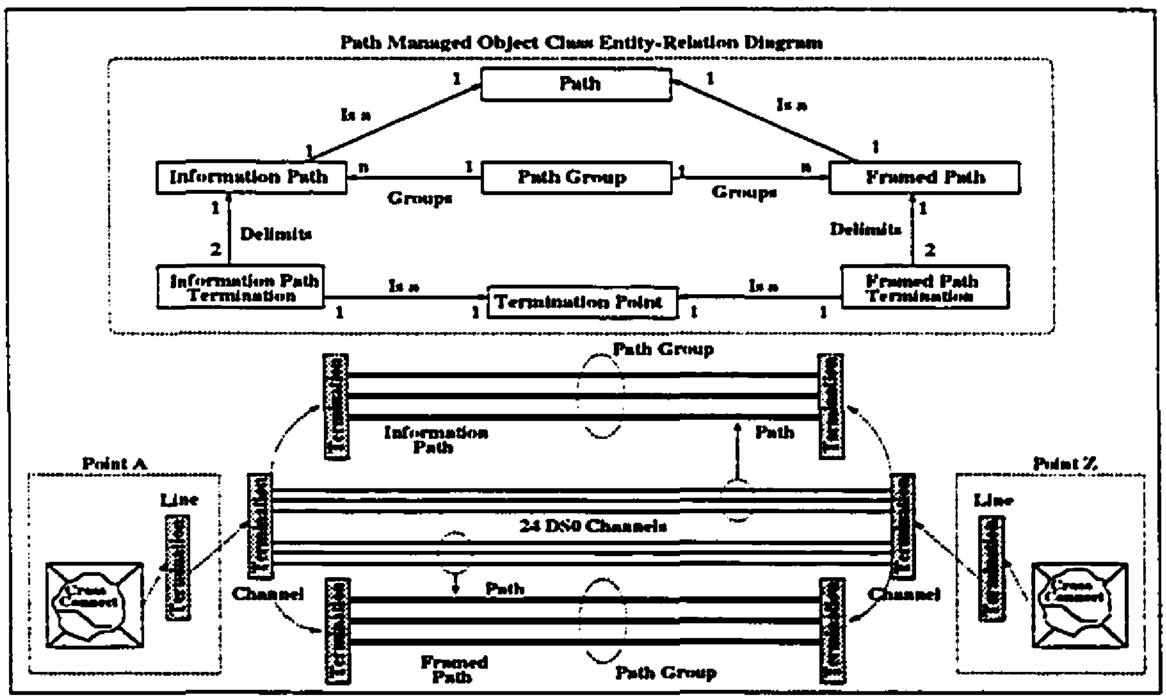

Figure 2.10: Path and Path Group Class-Entity Relationship

Figure 2.11 illustrates the association between paths and path groups. A path is created via a CreatePath method with source and destination (i.e., points A and Z) end-points and the path bandwidth (i.e, rate) and type (i.e., information or framed). Based on the requested rate, the CreatePath method scans termination points associated with the network element at point A for a channel of equal bandwidth. Since a channel is connected to adjacent network element termination points, the CreatePath method scans cross connections at the network element hopping from one channel to the next between network elements until it reaches the destination point Z. The list of channels interconnected between source and destination are the elements of the 'ChannelNameList' attribute of the *Path* managed object. The search procedure described above was programmed within the ODBMS schema; each channel at point A with bandwidth 'Rate' became a search thread for an end-to-end network path.

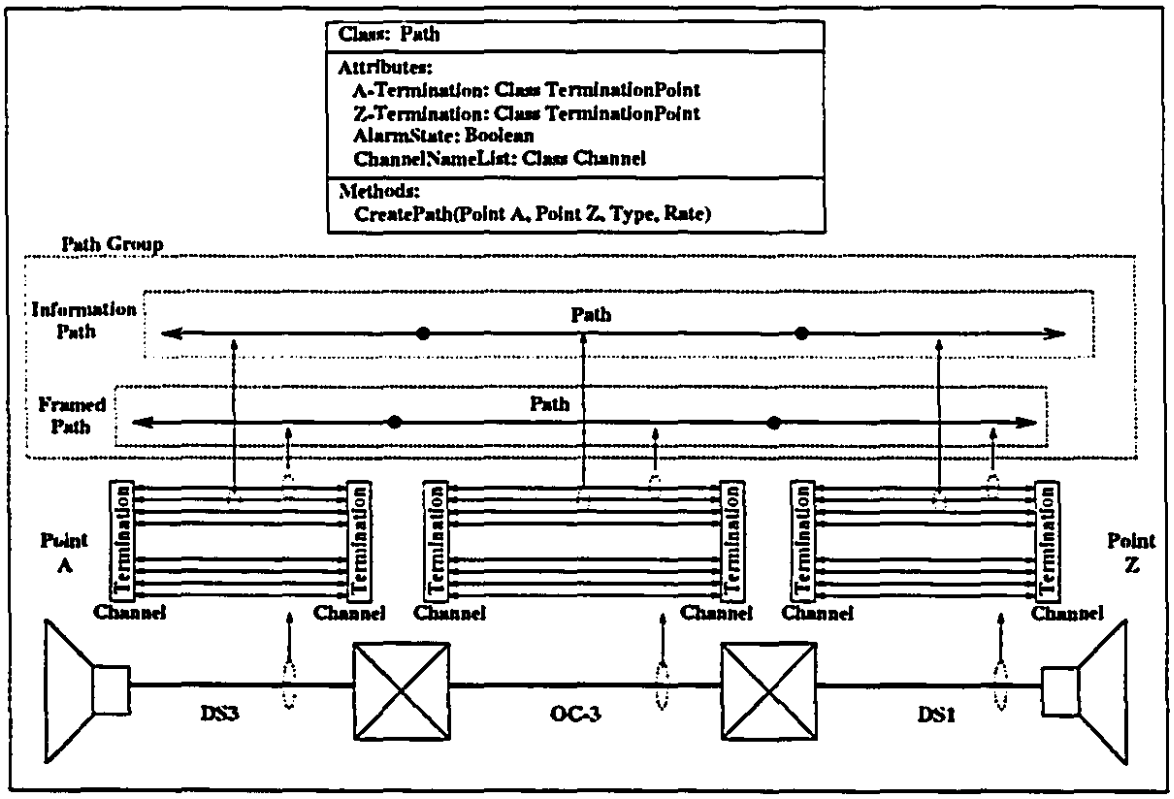

Figure 2.11: Path Group Instantiation and Behavior

It is important to note that our search procedure was selected for its simplicity; specifically, our search algorithm added to the *Path* managed object 'ChannelNameList' attribute, the first *Channel* managed object which matched the requested rate. However, many channels at a network element node can match the requested rate without terminating at the desired destination. As a result, in a distributed ODBMS environment containing hundreds of *Channel* managed objects, our search procedure might consume large amounts of computing resources before finding a suitable path between source and destination. In addition, since the newly created *Path* managed object is composed of the first sequence of channels which match the requested line rate, the path may not be optimal in cost of transmission.

Clearly, there is a need for some form of *look-ahead* procedure and optimization algorithm that can minimize the computing resources and search time required to create an end-to-end path. One possible approach is to use the routing tables associated with switching network elements. Routing tables can provide a list of possible communication paths between any two network elements based on the lines provisioned in the network. Through routing tables, we could associate a cost to each transmission line between any two network elements, and, this cost can be variable and dynamically allocated depending on usage and time of day. By implementing the routing tables as objects within our distributed ODBMS environment, we could use the concepts of object orientation to allocate cost to channels, paths and path groups. For example, a path created with channels derived from a line would inherit the transmission cost of the *Line* managed object. Furthermore, since both managed objects and routing tables would share a common ODBMS environment, they could communicate by passing messages through well defined interfaces. However, routing tables usually need to be updated very quickly when changes occur within the network; the ability to implement and maintain routing tables within an ODBMS environment would require further research.

2.3.5 Network and Network Services

Figure 2.12 illustrates the GNM **Network** and **Service** managed object class hierarchies and their entity-relationship diagram. A *Network* managed object represents a collection of interconnected telecommunication and management objects capable of exchanging information, and a network managed object has one or more common characteristics, for example, it may be owned by a single customer or provider, or associated with a specific service path. Furthermore, a network may be nested within another (sub)network, thereby forming a containment relationship. A *Service* managed object represents an offering or feature associated with a specific network functionality, and services may be nested within other services, thereby creating a containment relationship. Finally, a service is usually not stand-alone, but is embedded in an existing system of services. For our prototype, a service is provided via a path group composed of an aggregation of *Path* managed objects (i.e., information path or framed path) between information source and information sink.

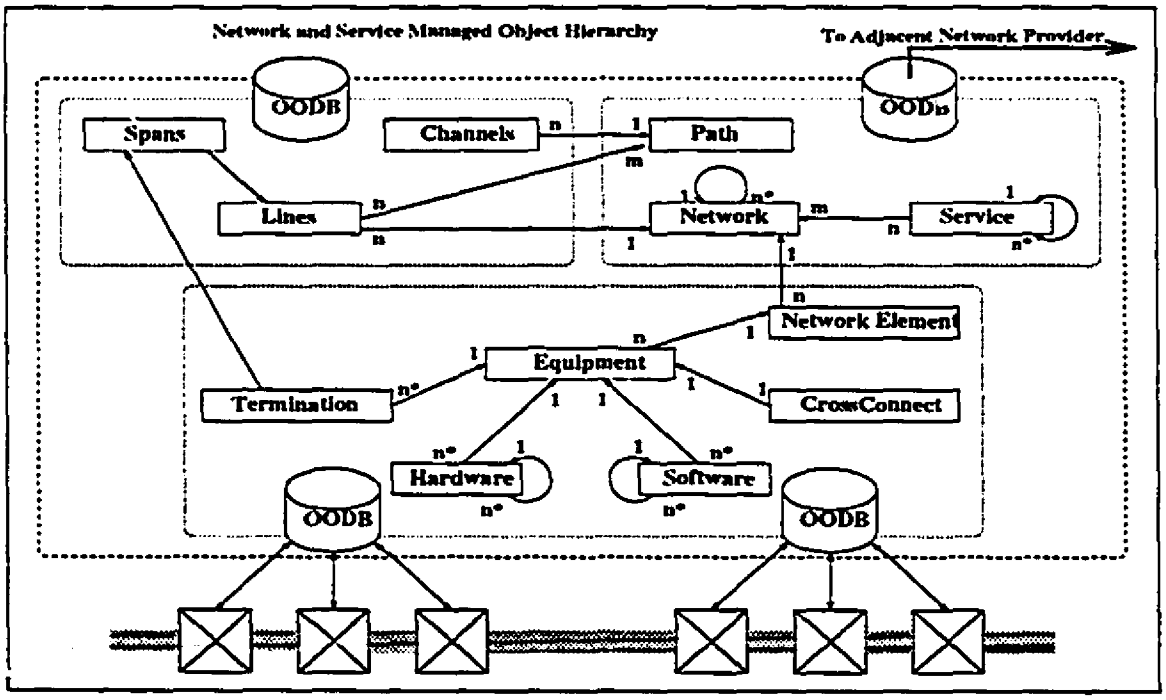

Figure 2.12: Network and Service Class-Entity Relationship

Figure 2.13 illustrates the association of *Network* and *Service* managed objects for telecommunications OAM&P. To begin, a *Network* managed object is instantiated by the CreateNetwork method with *Network Element* objects and *Line* objects as parameters. Since a line can carry many channels, several network services (i.e., path groups) can be provisioned through a common network. A network service is associated with the telecommunication (sub)network via an AssociateService method with a *PathGroup* managed object as parameter. Within our prototype, the AssociateService method is a function intended to calculate the service attributes of a given path group. Telecommunication service attributes allow users and management systems to determine the most appropriate network path for a given end-to-end service request, for example, the Quality of Service (QoS) attribute allow users and management systems to determine compatibility of connection between network domains.

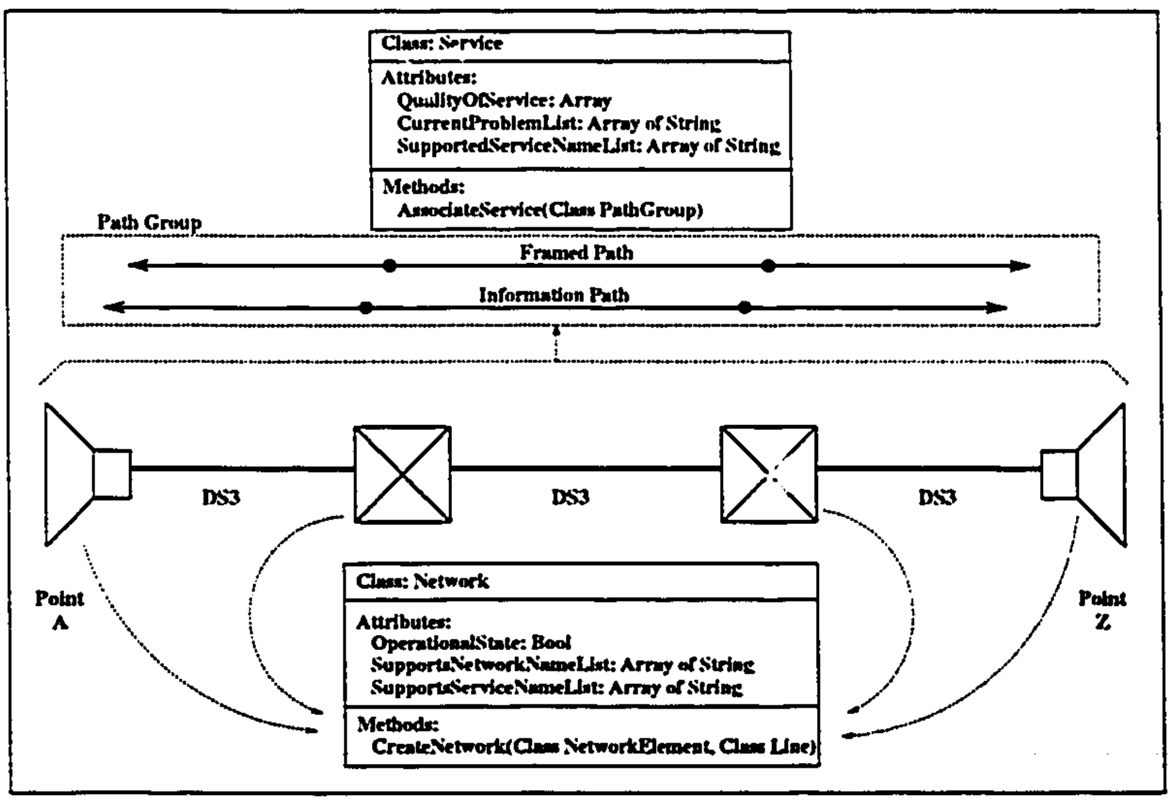

Figure 2.13: Networks and Services Instantiation

2.4 Conclusion

Our proposed network management architecture presents our view of a possible *integrated* network management environment for telecommunications OAM&P. Our network management view is built upon an *active* and *dynamic* standard network information model for interoperability between operations systems and network elements, and, distributed network management across multiple network domains. By *dynamic* we mean a MIM that can be extended to accommodate new managed objects associated with network resources, technologies and services introduced within the telecommunications network service domain. Specifically, we present a distributed network management environment where OODBs may be flexibly organized to reflect specific device configurations associated with a network element, while ODBMSs allow these different MIBs to be easily unified through a common database schema. As a result, a dynamic database can provide flexibility and efficiency in managed information access, where managing entities can control the content and structure of the MIB.

An active and dynamic MIB is very powerful because a human operator can define not only the rules to evaluate, but also when the evaluation has to be done. An active and dynamic MIM, however, does present significant challenges: first, the resources required to store and process managed information cannot be predicted at design time; second, changes in the ODBMS schema may result in corruption of distributed OODBs. Furthermore, it is almost impossible to avoid the side effects resulting from the execution of the triggered actions and the control of cascaded actions in the case of the violation of a rule is difficult to predict and maintain[58]. In order to understand these issues more fully, our next step is to implement a *network management prototype* testbed to further study the impact of *active* and *dynamic* MIBs for telecommunications OAM&P.

Chapter 3

Network Management Prototype

3.1 Introduction

We present a network management platform for telecommunications OAM&P, implemented within a lab environment[1]. Our prototype network management platform was used to evaluate and understand the capabilities and limitations of today's ODBMS technology as a repository and manager of an object-oriented MIM for distributed network management across multiple telecommunication network service provider domains. In particular, we want to investigate how an existing *off the shelf* ODBMS product can merge within a distributed object-oriented network management environment as a MIB of an overall distributed MIM for telecommunications OAM&P. Our network management platform also includes a prototype Graphical User Interface (GUI) for telecommunication network service provisioning. For our research in distributed network management, the prototype GUI was implemented to understand how graphical resource icons representing network elements and telecommunications OAM&P processes associated with the MIM can help human operators achieve a more efficient management of the telecommunications network. We believe a GUI can increase the human operator's productivity because of the simple-to-use input devices (i.e., icons, windows, and menus) which considerably improve communications between human users and computer systems.

[1]*Information Networks and Systems Laboratory*, McGill University, Montreal.

3.2 Network Management Prototype Setup

Figure 3.1 illustrates our prototype network management platform for telecommunications OAM&P. Here, we introduce *management information servers* composed of ITASCA Object Database Management Systems[2]; together, these ODBMS form a repository and manager of the prototype network information model presented earlier[3]. Furthermore, management information servers are connected to *management information browsers* capable of remote login sessions to the ODBMS environment, to monitor and change the database management information schema. Finally, the ODBMS environment is connected to *network management terminals* for telecommunications OAM&P through a human-machine interface. For our experiments, management information servers also simulated network element alarm *messages* in order to study the mechanics of *message passing* between managed objects.

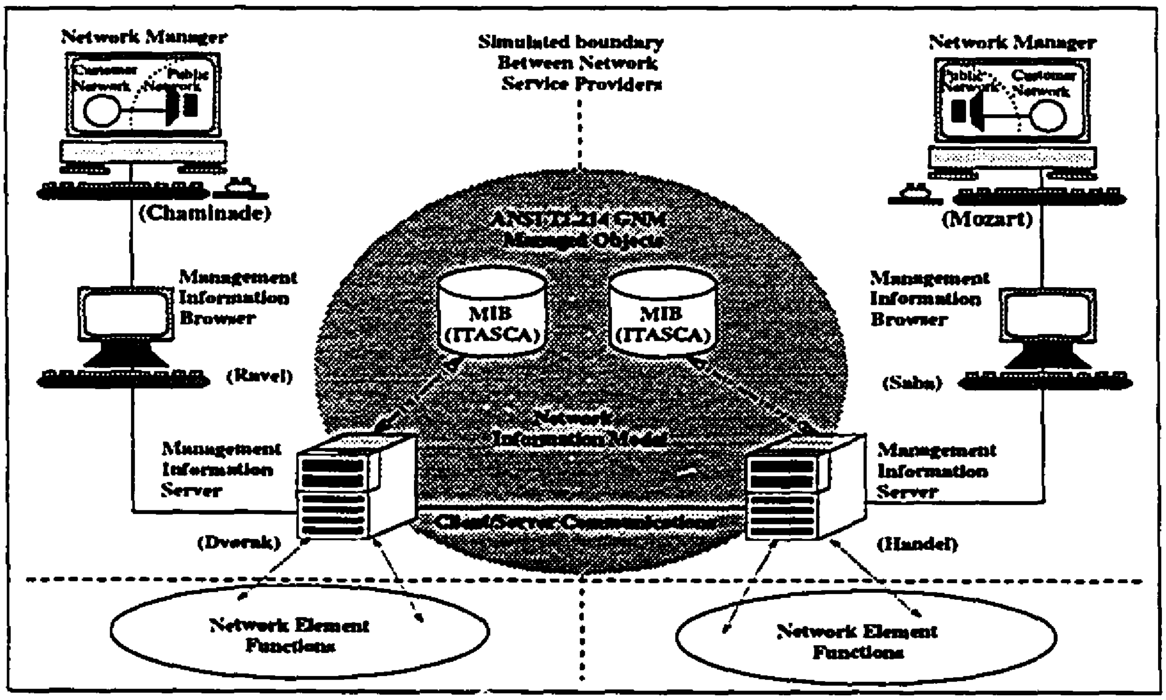

Figure 3.1: Distributed Network Management Simulation Setup

[2]ITASCA Object Database Management System is a trademark of Itasca Systems inc.
[3]The ITASCA ODBMS was selected based on the evaluation presented in [5] and [59].

3.2.1 ITASCA Object Database Management System

The ITASCA ODBMS is an extension of the ORION-2 prototype resulting from the ORION research project[60]. The research focus of the ORION prototype was to identify the impact of object-oriented concepts on the architecture of a database system and to validate solutions[54]. The ORION research uncovered several significant changes to the architecture of *conventional* database systems that object-oriented concepts necessitate; these changes include storage structures, queries, indexing, schema evolution, concurrency control, and access authorization.

The result of the ORION research is a data model that implements high level semantic modeling concepts such as multiple versions, including generic references and version derivation and composite objects with semantics for the treatment of composite object components. The ORION prototypes included a query optimizer similar to that found in relational systems, but modified for class hierarchies. In addition, the ORION prototypes introduced class hierarchy indexing, the implementation using a structure much like a B+ tree. The ORION prototypes extended the concepts of concurrency control with locks at the attribute and object level, to handle class hierarchies and dynamic schema evolution. The storage subsystem of the ORION prototypes implemented an *object-caching mechanism* for in-memory object management, in addition to the traditional *page caching* used by many database systems[61]. In-memory object management is possible because locking is at an *object level*; only the object is kept in memory, not the page on which it was stored. Finally, a taxonomy of *schema evolution* was developed as part of the ORION prototypes for adding or removing attributes and dynamically changing the class inheritance at any level in the hierarchy in a multi-user environment. The ORION prototype also supports multimedia objects such as text, images and audio as part of the class hierarchy and created much like any other object.

Figure 3.2[4] illustrates the ITASCA ODBMS, a distributed multi-client/server architecture with an ITASCA server controlling partitions of the shared database, while clients are provided transparent access to the various partitions of the shared database. ITASCA has neither a central data server nor a central name server, and, no single site acts as a master site; thus, ITASCA's architecture has no single point of failure[62]. The architecture allows any number of private and shared databases; here, private databases isolate data that is not shared with other users. Conversely, shared partitions allow different components to be distributed across the ODBMS environment. ITASCA stores each instance of the data in one site; however, the system or user may move the data from one site to another to improve data locality. As a result, there is no need for a user or application to know the specific location of the data, as ITASCA will automatically find the location, thus simplifying distributed application development and ensuring transparency of distributed transactions.

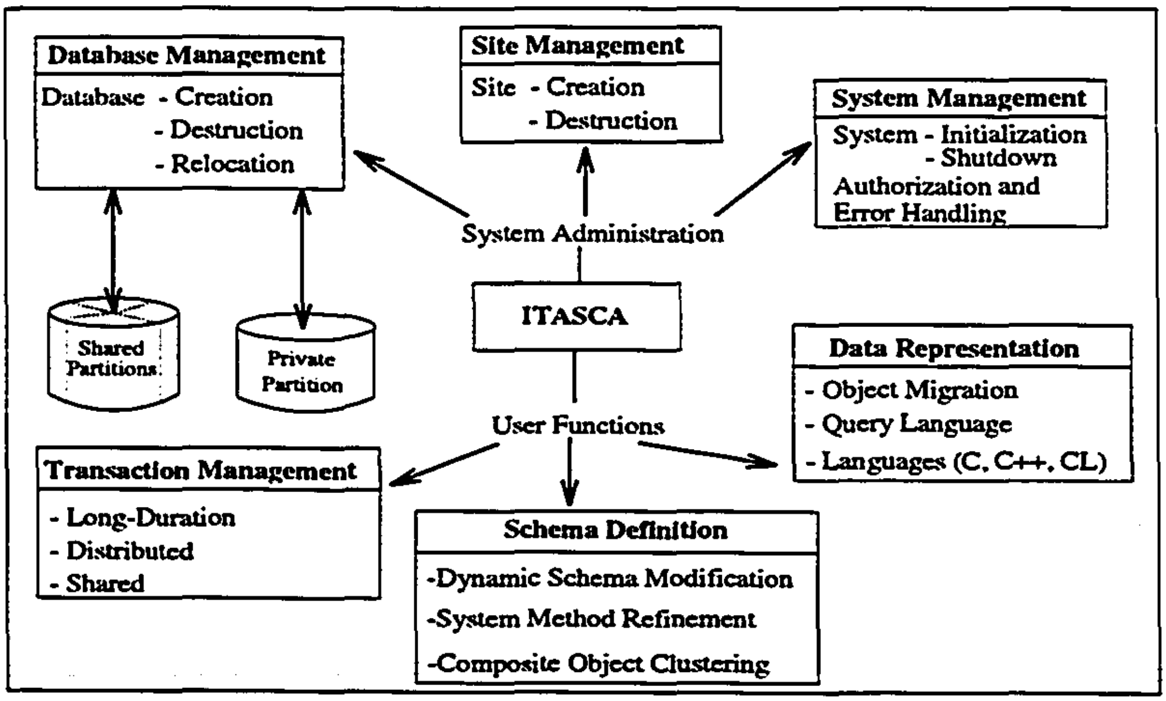

Figure 3.2: ITASCA Architecture

[4]Figure adapted from *Distributed Object Management Systems*[62].

Transactions guarantee the so-called ACID semantics. By ACID semantics we mean, *Atomicity*, *Concurrency*, *Isolation* and *Durability*: **Atomicity** means that an operation is only performed as a unit, either fully committed or aborted. **Consistency** means that a transaction transforms data from one consistent state into another. **Isolation** means that concurrent transactions execute as in a sequential system without interference. Finally, **Durability** means that the effects of a transaction remains persistent after completion[58][63].

ITASCA supports a *two-phase commit* protocol for distributed transactions: In phase 1 all transaction participants are polled as to whether they are able to successfully commit. In phase 2, the uniform decision is propagated in order to commit or abort jointly. Together, these phases ensure that every transaction is *atomic* and *durable*. In the case of concurrent transactions, ITASCA maintains a consistent state, preventing changes by one transaction from interacting in an uncontrolled manner with changes by another transaction[62]. In addition, ITASCA applies pessimistic concurrency control which implements serializability, for simultaneous, independent transactions to execute in parallel. Transactions request a lock on the object; a transaction may wait for a lock to become available until ITASCA detects a deadlock. Upon a deadlock, control is passed back to the user or application indicating that the last action was cancelled. The user/application can either re-execute or abort the transaction[64]. Finally, ITASCA introduces short and long duration transactions. A short duration transaction is a sequence of operations grouped into an indivisible *atomic* operation. If a transaction aborts, or fails, then none of the changes appear in the persistent database. Long duration transactions allow users to check objects out of the shared distributed database and into their private database. Users can then change the objects in the private databases without affecting the shared database or other users. These changes can be committed to the private database. Later, users can check the updated objects back into the shared database.

Figure 3.3 illustrates the partition of our prototype network information model within the ITASCA ODBMS environment. ITASCA has a sophisticated security authorization technique tied to the class hierarchy, and it supports both positive and negative authorization at any level in the class hierarchy. Authorization can be assigned to a class, an attribute, an instance, or a method. For example, granting access to all objects, but one, requires only two authorizations, a global access followed by a specific denial. Authorization can also be modified to exclude read authorization on a given instance through a negative authorization on that instance. Authorized users can add and remove attributes or change the subclass/superclass relationship at any time, and authorized users can also add or remove partitions of the shared database at any time. Authorization is also implicitly inherited along with the class hierarchy; as a result, inheritance of authorization reduces the work of database administration. All this can be interactively done while using the database without affecting other parts of the ITASCA database at the time changes occur to the schema.

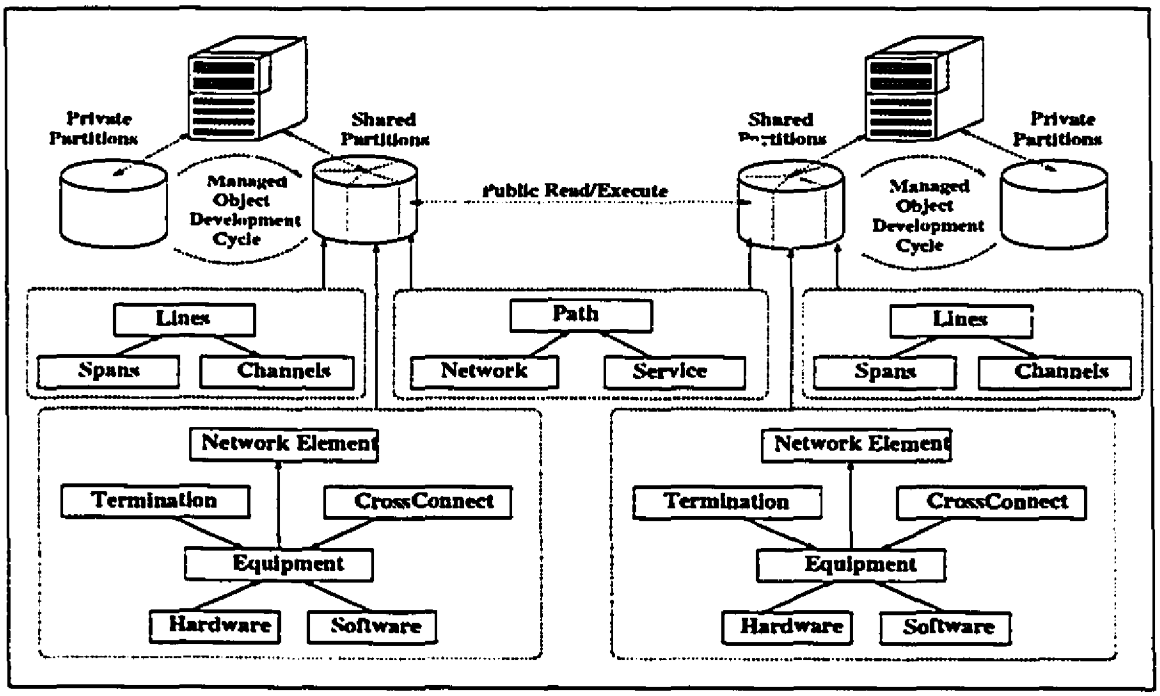

Figure 3.3: Information Model Partitions and Authorization

ITASCA supports a Motif[5] Schema Editor[65] for schema modification; its objective is to provide *evolution transparency* by assisting and maintaining changes to the database content. In ITASCA, modifications include simple changes to attributes or methods and complex tasks such as creating new sites, setting authorization, and, changing the inheritance hierarchy[60]. ITASCA supports dynamic schema modification resulting in a flexible environment for changing or customizing a database system. ITASCA redundantly stores the schema at each site to improve performance and evolution while management of schema updates is automatic for all sites, including sites that were offline during any changes.

Version control is one of the most important data-modeling requirements in OODB applications. After the initial creation of an object, new versions of the object are recursively derived from it. Versions give rise to at least two types of relationships: First, a *derived-from* relationship between a new version of an object and an old version of the object from which the new version was derived. Second, a *version-of* relationship between each version of an object and an abstract object that represents the object. Any number of new versions may be derived from any version at any time, and a version may be derived from more than one older version. Version support does incur storage and processing overhead and can seriously impact system performance when used extensively[61]. ITASCA supports *versions* of object instances; ITASCA maintains a *single version derivation hierarchy* for the shared database and all private databases, with each private database having a corresponding sub-hierarchy. A user may derive multiple versions from the same object; ITASCA supports both static and dynamic binding of references to versioned objects and ITASCA uses generic versions for dynamic binding. For schema integrity, ITASCA retains the version derivation hierarchy after deleting intermediate versions. Finally, ITASCA allows customized methods related to version control for specific application domains such as checking in/out versions.

[5] *Motif* is a registered trademark of the Open Software Foundation.

3.2.2 Prototype Human-Machine Interface

The purpose of our Human-Machine Interface (HMI) is to provide users with *management services* independent of the underlying network technology or network element vendor. By *users*, we mean network operators tasked with network management objectives, or remote telecommunication network service providers. While communication requirements for internal users can be well defined, requirements for external users can vary greatly, from service vendors requiring proprietary operations systems integration to end-users requiring user interface equipment on customer premises. Developing a detailed analysis of user requirements for OAM&P is the role of *requirements engineering*[6] and is beyond the scope of this research.

We view the HMI composed of workstations and GUIs. The workstation is intended to support user authorization and implement functions needed to provide user friendly facilities to enter, display and modify details about the network. The GUI provides the means to interpret and translate the network management information into a displayable format for human-machine interaction toward telecommunications OAM&P. The role of the GUI is to circumvent the syntactic barrier of heterogeneous workstations by presenting semantically related objects that allow users to operate with personally meaningful abstractions while ignoring the details of the computer[66]. A common management interface can provide users with the illusion of a single platform for telecommunications OAM&P without regard to the peculiarities and locations of the multiple network elements. Although it is desirable to maintain a consistent *look and feel* across applications, different users have different requirements for interface tools and different preferences for human-machine interaction. A GUI must, therefore, allow a broad range of interface styles and must be customizable on a per-user basis for optimal human-machine interaction.

[6]Requirements engineering is an activity of knowledge acquisition and formalization, in which customer expectations are investigated and documented; it must not only describe the current and future system and its goals, but also its domain and range of influence[53].

A GUI is inherently difficult to build without abstractions that simplify the implementation process; this difficulty is compounded by user demands for customizable user-interfaces. Objects are, however, a natural choice for representing the tools and elements that compose a user-interface and for supporting their direct manipulations. As a result, the object-oriented paradigm has proven successful in the presentation area, and the increasing hardware capabilities of workstations coupled with the object-oriented programming paradigm for graphical user interfaces have made available a powerful tool for interfacing users with systems. For example, Figure 3.4 illustrates our prototype network management GUI implemented using the InterViews toolkit and graphical user interface builder[7]; it is a front-end to our network management platform, where operations executed via on screen menu items are converted into messages communicated to the ODBMS environment.

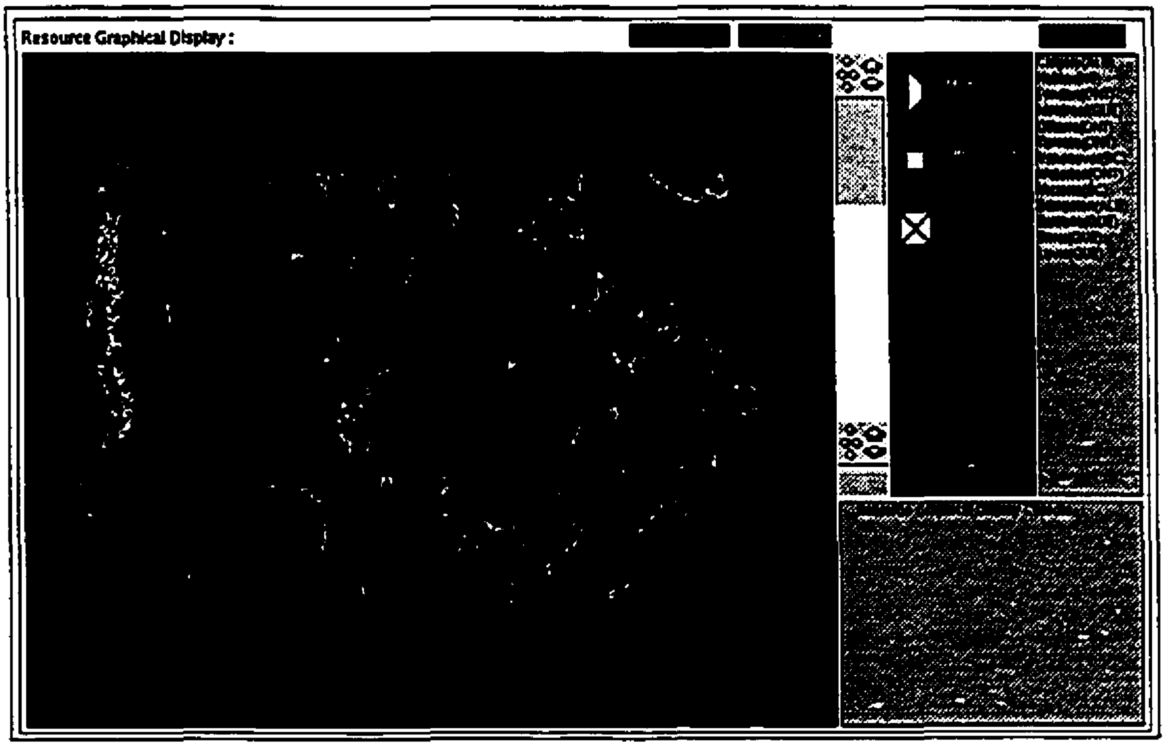

Figure 3.4: Prototype Graphical User Interface

[7]InterViews is a library of C++ classes that define common *interactive objects* and *common composition strategies* for graphical user interfaces[67].

Figure 3.5 illustrates the initial tools of our prototype GUI: *Console* window for system messages, *Online Help* window for manual pages describing the various components of the GUI, and *Login* window for user authentication. These tools provide a first level access for human-machine interaction with the prototype network management platform. In particular, the login window provides the means to resolve the *authorization level* of the human operator for access privileges to the MIM within the ODBMS environment. For example, access privileges are used to determine if a human operator is a *network designer* with authorization to modify the MIM to accommodate new network resources and services, or, a *management service user* with authorization to execute management processes enabling connections between network domains. In our prototype GUI, the *login process* helped define the users *problem domain* within the ODBMS environment; this information is then used to enable tools and semantic objects for telecommunications OAM&P and distributed network management across multiple network domains.

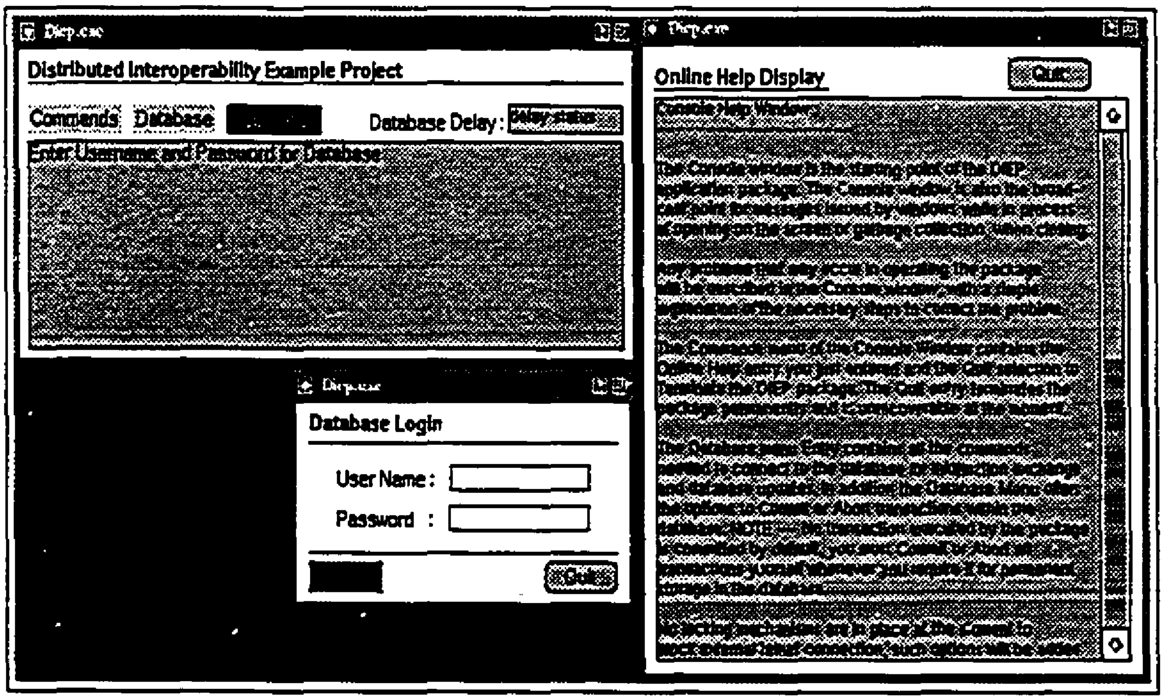

Figure 3.5: Top Level GUI Management Tools

Figure 3.6 illustrates a network designers *Network Element Creation* tool used to populate the ODBMS environment for telecommunication network service provisioning. Here, we highlight the use of graphical *resource icons* semantically related to the network element models presented in our prototype network information model. Through this tool, a network designer selects a resource icon and geographical location associated with the resource, and the GUI and ODBMS environment communicate to determine a set of default configuration parameters based on the resource and its network domain. Once the network designer has completed/validated the resource configuration, the data is submitted to the ODBMS environment to create the managed object. This ***automated*** approach to resource configuration can considerably help reduce human error in telecommunication management. Furthermore, managed objects in the ODBMS environment could be instantiated by an ***auto-discovery*** function following a boot process from a telecommunication resource in the network, while our proposed tool is used to customize tunable parameters such as trigger levels.

Figure 3.6: Network Element Creation Tool

Once a network designer has partitioned the telecommunications network into channels and paths, a customer representative can use a *Service Provisioning* tool, as illustrated in figure 3.7, to allocate path groups for end-to-end customer services. If the paths extend beyond the local network domain, the distributed ODBMS communicates with shared partitions of ODBMSs in adjacent network domains to enable paths through network access points between customer source and destination. However, the interaction between distributed OODBs and between managed objects are entirely hidden from the user by the human-machine interface. As such, the customer representative can focus on the problem domain of provisioning end-to-end customer services by providing the GUI with source, destination and the *quality of service* requested by the customer. If the network management environment can satisfy the request, the GUI illustrates the path group on screen and confirms it operational state; if the request is refused, the ODBMS environment sends a message to the console window describing the error.

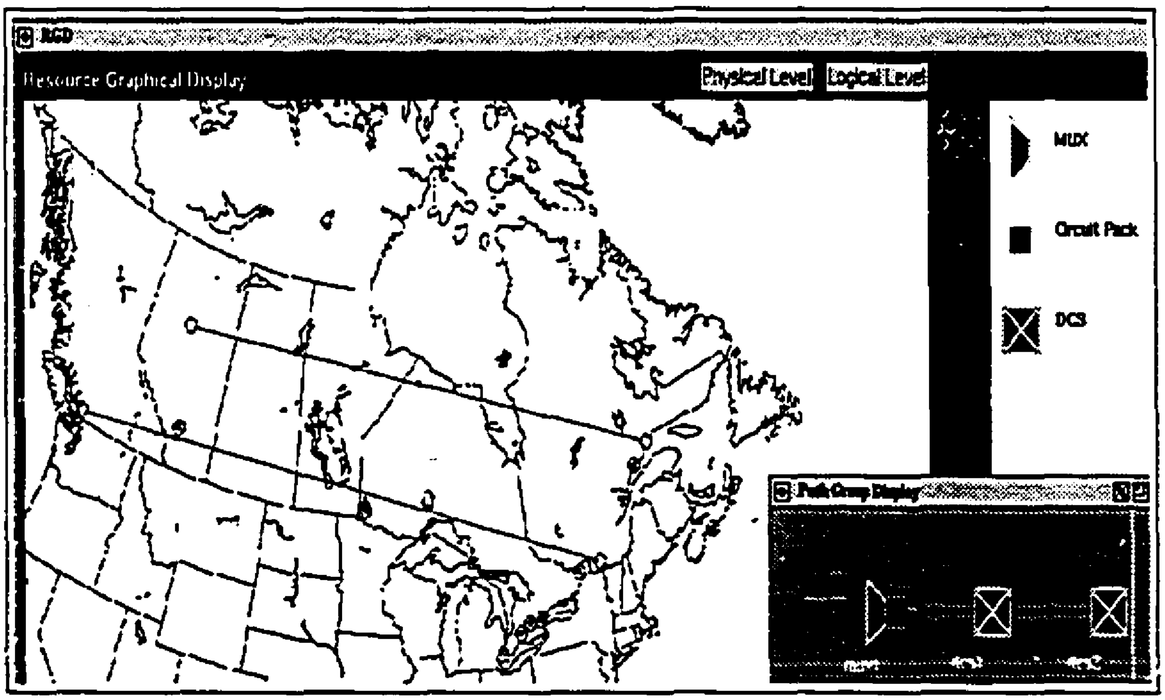

Figure 3.7: Telecommunication Network Service Provisioning

As an object-oriented environment, InterViews considerably helped in maximizing the reuse of software to quickly design new graphical user interface capabilities, as illustrated in Figure 3.8. In particular, InterViews supported the composition of *interactive objects* (i.e., buttons and menus), *structured graphic objects* (i.e, circles and polygons) and *structured text objects* (i.e., words and white space) in a toolkit implemented as a hierarchy composition of object classes derived from a common base class that defines the communication protocol for all objects in the hierarchy. Through inheritance, hierarchical composition gives the *interface builder* considerable flexibility, where a toolkit designer can concentrate on implementing the behavior of a specific component in isolation, while a graphical user interface designer is free to combine components in any way that suits the user's problem domain. Between the graphical user interface environment and ODBMS environment, objects and managed objects provide standard interfaces for a common communication protocol which provides quick and efficient software development.

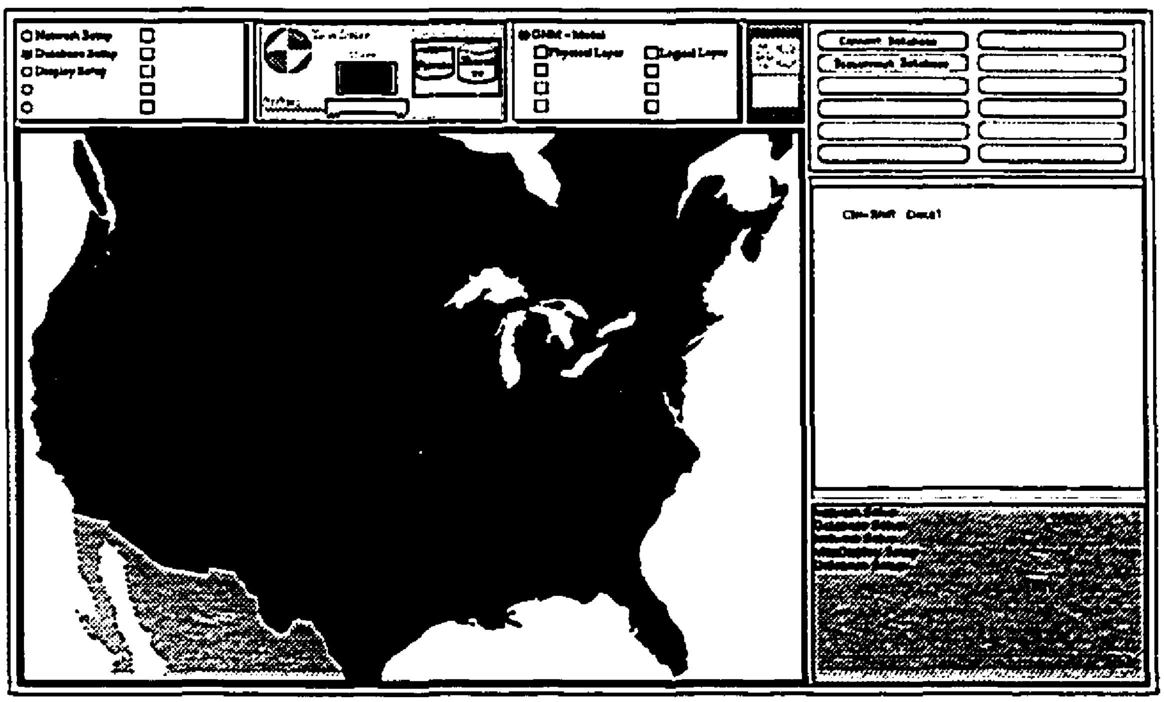

Figure 3.8: Software Reuse for Interface Development

3.3 Distributed Management Information

In our research on distributed network management, the ITASCA ODBMS provided a powerful environment for studying and understanding the benefits of a dynamic MIB over static databases for the provision, cessation and modification of network functions for end-to-end telecommunication services. Specifically, a static database structure may lead to difficulties in handling composite telecommunication device structures. Since different network components may require their own database models, they cannot be unified into a single MIB due to its static structure. ITASCA, on the other hand, provides a distributed and *dynamic* database structure, where each ODBMS shared a common and *active* MIM in the form of our prototype network information model. As telecommunications software designers, we could program intelligence within the ODBMS based on a common Management Information Tree (MIT) and MIM (i.e., managed object classes) of telecommunication resources, network and services. Changes to the MIM were automatically propagated to all OODBs within the ODBMS environment, with minimal user intervention.

The ITASCA ODBMS provides two forms of change notification: passive and active change notification. Passive change notification raises a flag, where a query can detect the change. On the other hand, active change notification acts as a trigger or daemon, with a change invoking the execution of a *notification method.* Because of this flexibility, the ITASCA ODBMS provides an excellent opportunity to evaluate the impact of poll-oriented management (i.e., passive notification) versus event-oriented (i.e., active notification) management. Poll-oriented management is used by the Simple Network Management Protocol (SNMP), a light-weight TCP/IP based protocol that is easier to implement, but less robust than CMIP. On the other hand, CMIP is an event driven management protocol with greater functionality than SNMP. Also, the SNMP management model is based on a Manager/Agent relationship in which the agent is simple and the manager is intelligent.

3.3.1 Active versus Passive Notification

To evaluate the impact of active versus passive notification, we implemented an event generator within the ITASCA ODBMS that randomly set managed object alarm attributes to simulate network element faults. Our objective is to measure the *average time delay* incurred by the ODBMS to provide timely management information at the GUI as the number of managed objects increases within the database, as illustrated in Figure 3.9. By **time delay** we mean the time elapsed between the moment an alarm is triggered and the moment the alarm data in provided at the GUI. In the *passive notification* scenario, the ODBMS schema was programmed to flag managed object alarms at the database interface. To collect these managed object alarms, a process external to the ODBMS environment was implemented to poll the database interface every sixty (60) seconds and query the flagged managed objects for management information. The sixty (60) second polling interval was selected to reflect the range of polling intervals found in SNMP network management today.

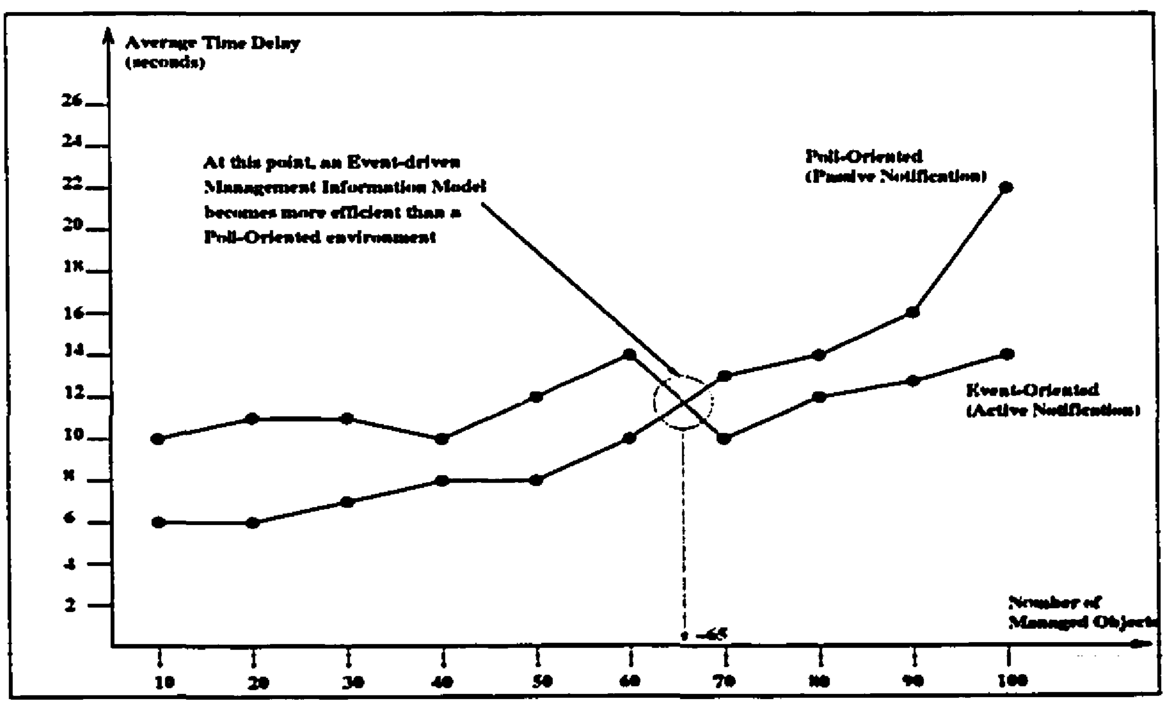

Figure 3.9: Performance of ITASCA ODBMS

In the *active notification* scenario, the ITASCA ODBMS schema was programmed to deliver management information at the database interface. In this case, when a event occurs within the ODBMS the counter measures the time delay to communicate the object alarm at the database interface. For both active and passive notifications, the time delay marked on the graph is the average value calculated from ten (10) measurements of time delay for each database size.

From our results, we clearly see that in a passive notification scheme, the time delay incurred by polling the ODBMS significantly increases as the number of managed objects grows within a central database. Furthermore, if we reduce the polling interval to below thirty (30) seconds we quickly saturate the database interface because the ITASCA ODBMS is queuing requests faster than it can resolve. Conversely, in an active notification scheme, the time delay incurred by the ITASCA ODBMS methods is approximately constant as the number of managed objects increases. In fact, we see a point at which an event-driven network management approach becomes much more efficient than a poll-oriented network management approach. Specifically, for a database size below this point, our active notification scheme incurred slightly more time delay due to the increased processing resources needed to execute the notification methods within the ITASCA ODBMS schema. For a database size above this point the passive notification scheme incurred an increased amount of time delay, to the point where the ITASCA ODBMS saturated with polling requests. Our experience with a passive notification Management Information Model leads us to believe that a poll-oriented network management approach must be carefully managed or it will consume large amounts of bandwidth in network management information transmission alone. This fact is not desirable in wide area networks and between multiple network domains where transmission costs can be high. Our next objective is to evaluate the scalability of the ITASCA ODBMS. By *scalable* we mean that the ITASCA database size can grow in size while maintaining its performance characteristics.

3.3.2 ITASCA ODBMS Scalability

To evaluate the scalability of the ITASCA ODBMS, we programmed the database schema as an event-driven MIM where network element managed objects were distributed across two ITASCA databases. As before, an event generator was used to simulate network element faults, and active notification methods were programmed into the database schema to communicate with the GUI when a fault occurred. Our objective is to measure the *average time delay* incurred by the **distributed** ODBMS environment to provide timely management information at the GUI, when management information is partitioned across multiple sites. The process was repeated for an increasing number of managed objects distributed in different proportions across both ITASCA ODBMS, as illustrated in Figure 3.10. Our goal is to evaluate the impact of distribution on management information processing and overall system stability and performance. Again, the time delay marked on the graph is the average value calculated from ten (10) measurements of time delay for each database size.

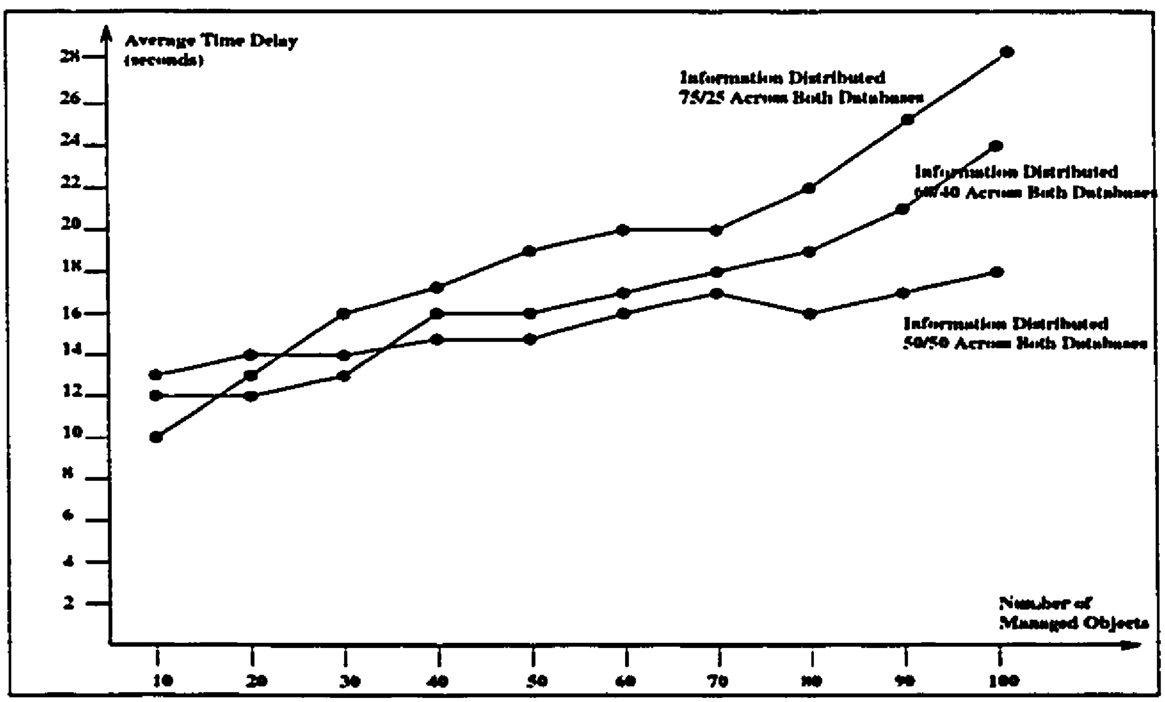

Figure 3.10: ITASCA ODBMS Scalability

Although our prototype network management platform was limited to two (2) ITASCA ODBMS, the benefits of data distribution was quite visible. Specifically, from our experimental results, we see that the average time delay for management information decreases as we distribute data more equally between both ODBMS. Furthermore, as the number of managed objects increases equally in both ODBMS, the increment in time delay is small and approximately linear since we are distributing the processing load between multiple workstation computing platforms. It should be noted, however, that in the distributed scenario, when most managed objects were located in one of the ODBMS, the time delay was significantly higher than in a single ITASCA ODBMS environment[8]. We believe this increased delay is the result of communications overhead between distributed ITASCA ODBMS coordinating database transactions across both nodes. Based on the data we have observed, we believe the time delays generated by the ITASCA ODBMS can be an obstacle for telecommunications network service providers reaching for real-time response. In particular, we see that as managed objects are distributed equally between both ITASCA ODBMS, the average time delay increased by approximately five (5) seconds while the ODBMS size increased to a hundred (100) managed objects. Since a telecommunications network can be composed of thousands of managed objects distributed across several MIBs, we believe the average time delays incurred by the distributed ODBMS environment would increase by an order of magnitude. It should be noted that our experiments with the ITASCA ODBMS were limited to a database size in the range of a hundred managed objects. For a database size much greater than a hundred managed objects, we observed that the ITASCA ODBMS would sometime lock-up during its *garbage collection* cycle. It was not possible to precisely identify the cause of this problem; however, we believe this could simply be a problem with our specific ITASCA product release (i.e., version 2.0) than with its object-oriented technology or implementation.

[8]In reference to the *active notification* data illustrated in Figure 3.9.

3.4 Toward Multiple Management Views

Our prototype network management platform builds upon the premise that all ODBMS share a *single and common* database schema. Specifically, each ODBMS contains a complete copy of the MIM, including resource, network and service managed object class definitions and notification methods. As a result, the MIB becomes increasingly resource intensive as its MIM grows in complexity, thus placing a large processing burden on NEs. This can be a problem for memory-constrained NE devices. Instead, a more efficient approach is to program the MIB with a standard MIM associated with a subset of network elements, as illustrated in Figure 3.11. Each MIB offers a *network element view* of managed objects and notification methods applicable to the subset of NEs. Since the MIB contains an *active* and *dynamic* MIM, management and notification methods can be programmed to communicate with telecommunications OAM&P processes residing at a higher operation systems layer.

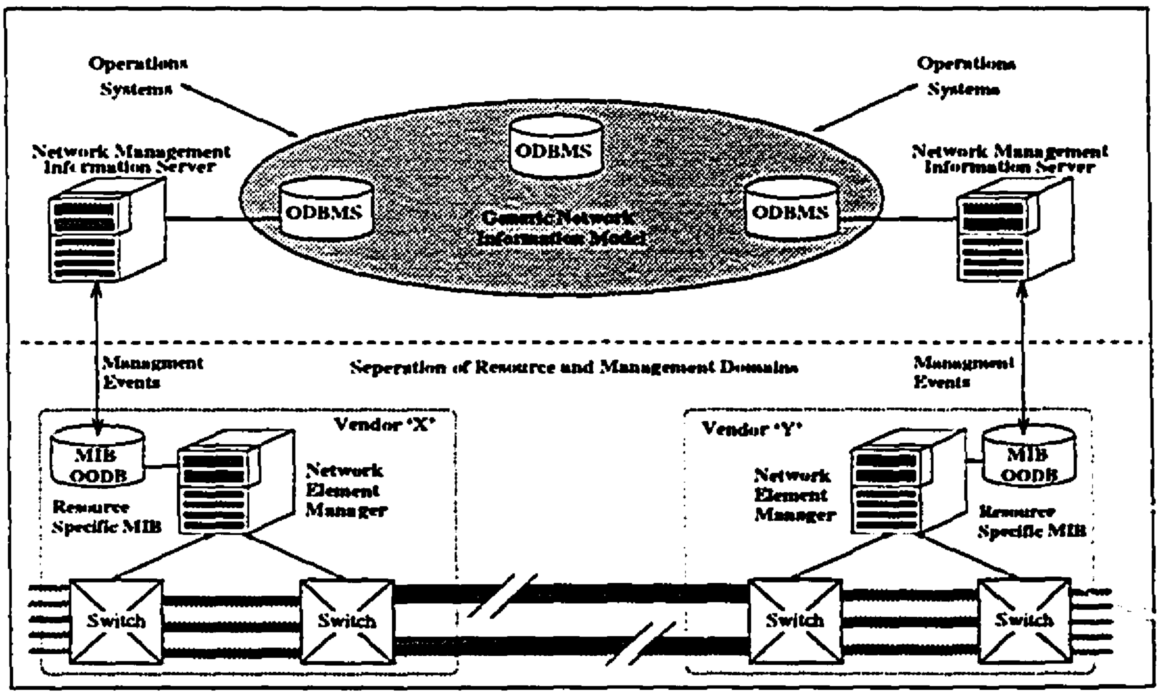

Figure 3.11: Separation of Network Element and Network Management Domain

This segmentation of operations systems can be extended to the entire operations infrastructure in the form of a Network Element Management Layer (NEML), Network Management Layer (NML) and Service Management Layer (SML), as illustrated in Figure 3.12. Here, each layer builds upon an ODBMS environment which contains a standard MIM (i.e, management view) of the telecommunications network derived from a common generic network model, and each management view is an abstraction of the immediate lower layer. For example, the NEML provides network management operations systems with a *vendor-independent view* of the network elements that compose the telecommunications network, and, the NML provides service management operation systems with a *technology-independent view* of the telecommunications network. The SML provides customers and work center staff with a *network-independent view* of the telecommunication services. Finally, we view telecommunication operations infrastructures communicating via these standard MIMs for end-to-end information networking across multiple network domains.

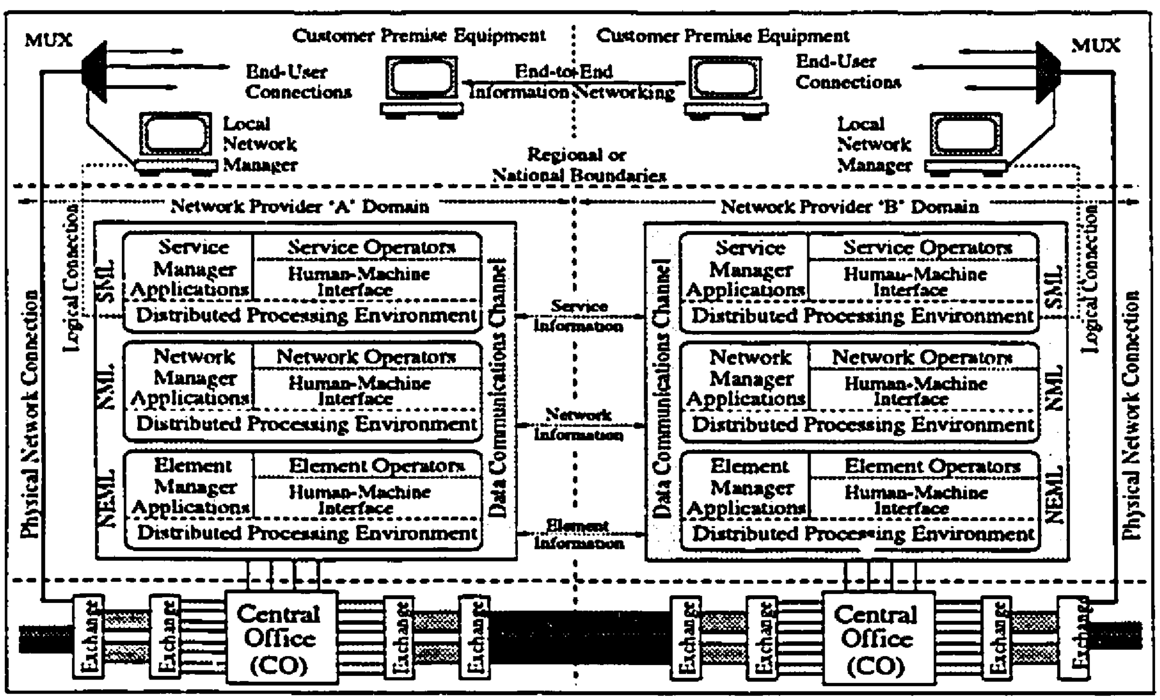

Figure 3.12: Distributed Telecommunications Architecture

3.5 Conclusion

The TMN M.3100 Generic Network Model (GNM) proposes several different viewpoints of management information[50]: (1) Network Element Viewpoint is concerned with information that is required to manage network elements individually or as a set. This refers to information required to manage network element functions and physical aspects of the network element. (2) Network viewpoint is concerned with the information representing the network, both physically and logically. It is concerned with how network element entities are related, topologically interconnected, and configured to provide and maintain end-to-end connectivity. (3) Service Viewpoint is concerned with how network level aspects are utilized to provide a network service and, as such, is concerned with requirements of a network service and how these requirements are met. The definition of *viewpoint* is a means of generating requirements for open interfaces; these viewpoints are not restrictive, rather, they define the levels of abstraction of particular types of interfaces. There is no implicit definition of interfaces or storage requirements; objects defined for a given viewpoint may be used in others, and any object may be used by any interface which requires it. Furthermore, the M.3100 GNM describes managed object classes and their properties that are generic and useful for information exchanged across all interfaces defined by the TMN architecture. In particular, M.3100 GNM identifies object classes that are common to managed telecommunications networks, or are of the generic type that can be used to manage a network at a *technology-independent* level, or are a super-class of technology-specific managed objects in a telecommunications network, or management support objects that are required for the management of the telecommunications network. By introducing the concept of technology-independent management, it is possible to perform management across different network technologies, architectures and services using common communication interfaces by enforcing an abstract view over a set of network elements.

Chapter 4

Future Research Directions

4.1 Introduction

An important lesson that the telecommunications industry must retain from the past evolution of telecommunication network service provider operations infrastructure is the reminder to always look at the big picture - to think about the operations infrastructure as a ***networked domain*** of interrelated customer and service provider resources combined in a vision for global partnership. At the heart of such global partnership are harmonized regional, national and global telecommunication standards; however, the telecommunication industry will also need new network and systems technologies capable of meeting the challenges for global networking[68]. In particular, with networks and services growing in a distributed nature, telecommunications networks are increasingly using computers and computers are increasingly being interconnected by telecommunications networks. As a result, distributed system technologies are becoming essential to develop and deploy new cost effective management and telecommunication services. In addition, software will play a major role in deploying new telecommunication services, and, interoperability, portability, and reuse of software components will become increasingly critical for efficient end-to-end information networking. Finally, for object based service and management systems there is no commonly accepted object model for modeling telecommunication services; however, ROSA (RACE Open Services Architecture) has developed an object model, the ROSA object-oriented model[69].

4.2 Distributed Systems Technology

Although the ITASCA ODBMS provided a feature-rich repository and management environment for network information modeling and telecommunication management methods, today's ODBMS technology still faces important challenges. To begin, there is no agreement on standardization within the realm of object orientation. Specifically, neither the boundaries for the query model have been set up nor has an object oriented query language been well defined yet. As a consequence, there is also no standard object oriented data model[54][70]; this is one of the common complaints against OODBs and distributed systems implemented using this technology.

Distributed system technology has become a major focus in international standardization; for example, work in ISO is in progress to define a reference model for Open Distributed Processing (ODP). The reference model will include descriptive, as well as prescriptive, parts. The descriptive part defines terminology and modeling tools that can be used to model arbitrary distributed systems. The prescriptive part specifies when a distributed system may be called an ODP system. Specifically, the reference model prescribes architectural properties that an ODP system must have. After the ODP reference model is completed, individual ODP standards conforming to the reference model will be defined. Most likely, one will first work on standards for infrastructure components similar to those that are found in the OSF Distributed Computing Environment (DCE) today[57]. Although ODP and DCE are completely unrelated, the ODP work on an abstract reference model can significantly benefit from the design of an architecture such as DCE. The latter shows what functionality is needed in distributed processing systems and how components can be integrated into a common framework. Furthermore, when individual ODP standards will be sought for, the OSF DCE technology should be a suitable and promising starting point. Finally, distributed object-oriented systems will also require standardization in the management of objects across platforms; work is underway in this area[71].

4.3 Distributed Object-Oriented Systems

The industry consortium Object Management Group (OMG) has defined the Object Management Architecture (OMA) for managing objects in distributed systems. OSF and OMG have expressed their interest in advancing the ODP standardization. This approach aims at providing support for distributed object interaction in a heterogeneous environment. In an OMA, objects usually tend to be much larger than they are at programming level (i.e., a whole program can be an object). This approach differs from object-oriented database technology in the sense that these *coarse-grained objects* are treated as monoliths. A key component of OMA is the Object Service Architecture that offers the required services with a very broad spectrum of functionality. In general, these services provide a higher level of abstraction than DCE does and cover a broader technological area, such as database and transaction-oriented services, version control of software objects, concurrency control, and distributed object replication. Location independent object interactions are supported by the Common Object Request Broker Architecture (CORBA); it supports mechanisms for identifying, locating, and accessing objects in a distributed environment. The OMA has not yet reached the same level of maturity that DCE has. Moreover, some functionality of distributed object-oriented systems mentioned above, namely mobility, is not yet supported by CORBA.

Furthermore, it is becoming essential for the telecommunications industry to adopt a common architecture which integrates information service applications and information management. As voice services and broadband services converge toward an integrated product, it will become expense and time consuming to maintain and evolve separate architectures, such as IN and TMN. Hence, OMA and DCE can help achieve an information networking architecture for a broad range of telecommunication and management services; such an architecture would embrace the IN and TMN architectures within a framework based on distributed processing principles.

4.4 Information Networking Architecture

The Telecommunications Information Networking Architecture Consortium (TINA-C) views an information networking architecture as a *distributed telecommunications environment* that provides customers the capability of on demand access and management of information[11]. The TINA *vision* embodies a distributed application platform for building and executing network wide applications; it's purpose is to enable distributed processing as opposed to centralized nodes, and, to eliminate the divisions between network applications and operations applications. The TINA Architecture builds on current advances in broadband communication and distributed computing technologies, specifying a software based architecture for future information networks that are required to transport multimedia communication. The TINA architecture prescribes an object-oriented model for development of distributed applications in a distributed processing environment (DPE), as illustrated in Figure 4.1.

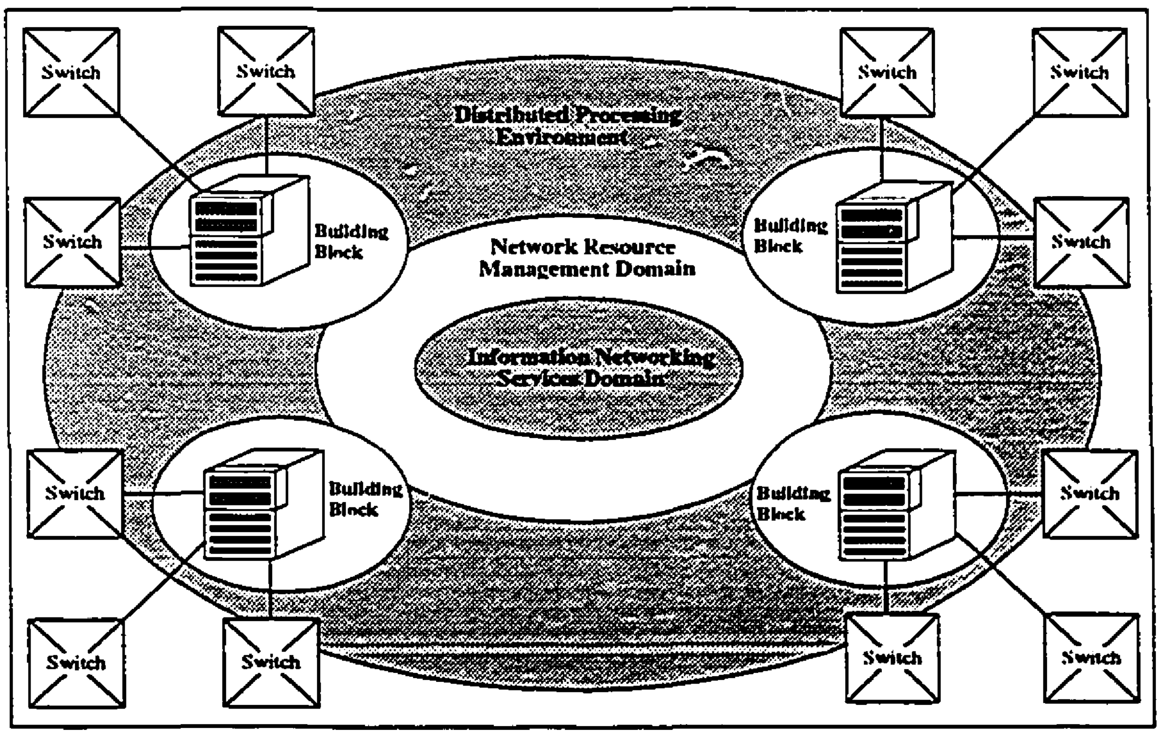

Figure 4.1: Telecommunications Information Networking Architecture

Chapter 5

Conclusion

Our research in prototyping an object-oriented approach for distributed network management has highlighted several benefits with a distributed ODBMS environment and with object orientation in general. To begin, we observed that the object-oriented paradigm provided features that helped maintain short development cycles (i.e., prototype, test and validate code) while building our network management GUI and network information model for telecommunications OAM&P. In particular, by applying the sound and proven principles of Object-Oriented Analysis (OOA) and Object-Oriented Design (OOD)[26][61] to the *problem domain* of distributed network management, we could abstract from the technical details of complex network elements and quickly identify management objects and their relationships for telecommunications OAM&P, as an extension of the ANSI T1.214 GNM. Concurrently, by applying the OSI standards for the structure of management information, to our prototype network information model, we could begin to define interoperability interfaces between managed objects during the initial stages of our research. Furthermore, the object model features of inheritance and allomorphism allowed us to refine the behavior of our prototype managed objects as we developed more generic network and service managed objects for telecommunications network management. However, for telecommunications OAM&P, it is the distributed ODBMS environment that provided the unifying platform for network management across multiple domains.

5.1 Advantages of a Distributed ODBMS

In a distributed ODBMS environment, object data representation is highly flexible and may be customized by users with little restrictions. For telecommunication network management, an ODBMS allows data to be polymorphic and intelligent. By polymorphic we refer to the concept that a name may denote managed objects of many different object classes and these managed objects may respond to some common set of methods in different ways. As a result, polymorphism plays a central role in refining managed objects within the ODBMS environment. In addition, the facilities of inheritance and schema evolution allow the network information model to grow incrementally. Furthermore, generalization and inheritance allow the management information schema to be better structured, more intuitive and to capture the semantics of the managed object environment. An important feature of the ITASCA ODBMS for distributed network management is the ability to store and activate methods directly in the database; as a result, the schema of our prototype management information model is language neutral. By *language neutral* we mean that managed objects stored using one programming language, can be accessed by other programming languages. For example, throughout our MIB development cycle, the management information schema was implemented using Common Lisp via the basic ITASCA ODBMS interface, while our GUI and telecommunication OAM&P test processes would communicate with the ITASCA database through a C/C++ interface. As a result, existing management applications, residing on operations systems, do not need to be written in an object-oriented language to communicate with the ODBMS environment. In addition, because of its object-oriented environment, it is possible to program extensive notification and management methods (i.e., *intelligence*) within the management information schema allowing external telecommunications OAM&P processes to perform more operations in fewer instructions, thus creating less network traffic between management systems and the ODBMS environment.

Distributed ODBMS are *identity-based* systems that can relate two or more objects independently of their embedded values. *Identity* is an essential feature of an ODBMS and enables the notion of sharing by having references from multiple locations in the distributed environment point to the same object, rather than maintaining copies at multiple sites. However, the ITASCA ODBMS environment did highlight an important problem with object identity. A managed object, when deleted from the ODBMS, will leave a *dangling object* pointer in a managed object attribute that was referencing the managed object. This 'dangling object' pointer would terminate (e.g., core dump) our processes because of invalid pointer operations on the management information. A simple work around was to implement a filter method in our prototype network information model that would eliminate these pointers when extracting management information from the database. However, we believe the ODBMS environment should eliminate these pointers from the database as part of its *garbage collection* function, and not through user defined methods. Finally, an ODBMS environment can provide data independence and location transparency. The benefits of location transparency and data independence are significant: first, maintenance costs are reduced because application source code need not be modified when objects are relocated; second, performance is enhanced because objects may be moved to ensure locality of reference and balance workloads; and, third, data integrity is preserved because object identifiers are logical not physical, therefore independent of location and do not change when objects are moved. Furthermore, from our experiments with active and passive notifications, we see that an active ODBMS schema is more suitable for today's growing telecommunication networks. In particular, since the ODBMS can notify on selected events determined by the network designer, communication costs between management systems and the network can be controlled and balanced. A passive notification environment, such as SNMP, may be acceptable for relatively small telecommunication networks with few network elements; however, implementation and operating costs quickly grow as network size increases.

Glossary

Agent:

An Agent is part of a network element application which executes management directives and emit notifications.

Allomorphism:

Refers to the ability of an object instance that is a member of one class to be managed as an instance of an object of one or several other object classes.

ASN.1: *Abstract Syntax Notation 1*

ASN.1 is a specification language which provides an orderly and clear method for assigning meaning to a collection of data values.

BER: *Basic Encoding Rules*

BER specifies an orderly data communications transfer and clear data descriptions.

CMIP: *Common Management Information Protocol*

Describes how to exchange basic management information between open distributed systems.

CMIS: *Common Management Information Services*

Provides a logical view of the management information to be exchanged between open distributed systems.

DMI: *Definition of Management Information*

Defines management support objects and generic attributes which are required by standards.

GDMO: *Guidelines for the Definition of Managed Objects*

Instructs implementors as to what a specific managed object should look like.

GNM: *Generic Network Model*

A management information model defined to support telecommunications OAM&P functions and data communication between operations systems and heterogeneous network elements.

IN: *Intelligent Network*

A generic architecture intended for various real-time connection and management services

INCM: *Intelligent Network Conceptual Model*

A framework for the design and description of IN Capability Sets and the IN architecture, for a particular phase of IN evolution.

Manager:

A Manager is part of a management application that issues management operations and receives notifications to monitor and control managed objects.

MIB: *Management Information Base*

It is a database that is shared between Managers and Agents to provide information about the managed network; it provides the structure and state of the network elements that a Manager operates on and an Agent enforces, abstracting from implementation and technological details.

MIM: *Management Information Model*

Defines how management data must be structured, including managed objects and attributes; what operations can be executed; and, what notifications are required.

OAM&P: *Operations, Administration, Maintenance* and *Provisioning*

A distributed information processing activity that supports the operations, administration, maintenance, and provisioning of telecommunications network services, and that is characterized by the cooperation of two or more OAM&P processes located in different real open systems.

OSI: *Open Systems Interconnect*

A Reference to protocols defined by the *International Standardization Organization* for interconnection of cooperative computing systems.

TMN: *Telecommunication Management Network*

Refers to a conceptually separate network that interfaces a telecommunications network at several points to send/receive information to/from it and control its operations.

Bibliography

[1] R. Sundstrom et al. SNA: Current Requirements and Directions. *IBM Syst. J.*, 26(1), 1987.

[2] Maurizio Decina. Network Evolution Towards Universal Personal Communications. In *Networks Without Bounds*, volume 1 of *Network Operations and Management Symposium*, pages 0026–0035. IEEE Communications Society, 1992.

[3] J. J. Garrahan, P. A. Russo, K. Kitami, and R. Kung. Intelligent Network Overview. *IEEE Communications Magazine*, 31(3):30–36, March 1993.

[4] Stephen Chen, Masanobu Fujioka, and Gerard O'Reilly. Intelligent Networking for the Global Market. *IEEE Communications Magazine*, 31(3):86–92, March 1993.

[5] B. Ellacott, S. Torrente and S. D. Morgera. The Object-Oriented Paradigm in a Distributed Network Management Environment. McGill University Report for Bell Canada, October 1991.

[6] Salvatore Torrente, Salvatore D. Morgera, and K. Wayne Lester. Implementation of the ANSI T1M1.5 GNM-T1.214 within an ODBMS Framework. *IEEE Computer Society – Networking: Foundation for the Future*, pages 875–884, 1993.

[7] H. M. Shooshan. *Disconnecting Bell : The impact of the AT&T divestiture.* Pergamon Press, New York, 1984.

[8] Robert L. Martin. Expanding the Boundaries of Network Operations. In *Networks Without Bounds*, volume 1 of *IEEE 1992 Network Operations and Management Symposium*, pages 0003–0014. IEEE Communications Society, 1992.

[9] William E. Gilbert. The Five Challenges of Managing Global Networks. *IEEE Communications Magazine*, 30(10):78–82, October 1992.

[10] P.E. Prozeller. Applying Distributed Systems Principles and Technology to Downsize and Improve Common Carrier Operational Infrastructures. In *Networks Without Bounds*, volume 3 of *Network Operations and Management Symposium*, pages 658–669. IEEE Communications Society, 1992.

[11] William J. Barr, Trevor Boyd, and Yui Inoue. The TINA Initiative. *IEEE Communications Magazine*, 31(3):70–76, March 1993.

[12] Michael A Hall. Integration of Operations Systems - Basic Strategies. In *Networks Without Bounds*, volume 3 of *IEEE 1992 Network Operations and Management Symposium*, pages 647–657. IEEE Communications Society, 1992.

[13] R. McCormick. Re-Inventing the Telephone Industry. *IEEE Communications Magazine*, 31(12):24–26, December 1993.

[14] Tokuo Lida. Domestic Standards in a Changing World. *IEEE Communications Magazine*, 32(1):46–49, January 1994.

[15] Theodor Ilmer. Shaping Future Telecommunications: The Challenge of Global Standardization. *IEEE Communications Magazine*, 32(1):20–28, January 1994.

[16] Robert E. Allen. A View of Divestiture, Ten Years Later. *IEEE Communications Magazine*, 31(12):18–23, December 1993.

[17] Decker Anstrom. The TCI/Bell Atlantic Merger: A Step Toward Telecommunication Competition. *Telecommunications - State of the Art*, pages 51–52, January 1994.

[18] Cesare Mossotto. Pathways for Telecommunications: A European Outlook. *IEEE Communications Magazine*, 31(8):52–58, August 1993.

[19] Gerald H. Peterson and Charles A. Dvorak. Global Standards. *IEEE Communications Magazine*, 32(1):68–70, January 1994.

[20] Arthur K. Reilly. A U.S. Perspective on Standards Development. *IEEE Communications Magazine*, 32(1):30–36, January 1994.

[21] Robert M. Amy. Standards by Consensus. *IEEE Communications Magazine*, 32(1):52–55, January 1994.

[22] Donovan Nak. Coordinating Global Standards and Market Demands. *IEEE Communications Magazine*, 32(1):72–75, January 1994.

[23] American National Standards for Telecommunications. *Operations, Administration, Maintenance and Provisioning (OAM&P) - Generic Network Model for Interfaces between Operations Systems and Network Elements*. Number T1.214 in American National Standards for Telecommunications. American National Standards Institute, 1990.

[24] The International Telegraph and Telephone Consultative Committee. Management Framework for Open Systems Interconnection (OSI) For CCITT Applications. Technical Report X.700, International Telecommunications Union, September 1992.

[25] S. Mark Klerer. System Management Information Modeling. *IEEE Communications Magazine*, 31(5):38–44, May 1993.

[26] G. Booch. *Object Oriented Design with Applications*. The Benjamin/Cummings Publishing Company Inc., Redwood City, California, 1991.

[27] The International Telegraph and Telephone Consultative Committee. Information Technology - Open Systems Interconnection - Systems Management

Overview. Technical Report X.701, International Telecommunications Union, 1992.

[28] The International Telegraph and Telephone Consultative Committee. Information Technology - Open Systems Interconnection - Systems Management: State Management Function. Technical Report X.731, International Telecommunications Union, 1992.

[29] The International Telegraph and Telephone Consultative Committee. Information Technology - Open Systems Interconnection - Systems Management: Attributes for Representing Relationships. Technical Report X.732, International Telecommunications Union, 1992.

[30] The International Telegraph and Telephone Consultative Committee. Information Technology - Open Systems Interconnection - Systems Management: Event Management Function. Technical Report X.734, International Telecommunications Union, 1992.

[31] The International Telegraph and Telephone Consultative Committee. Information Technology - Open Systems Interconnection - Systems Management: Alarm Reporting Function. Technical Report X.733, International Telecommunications Union, 1992.

[32] The International Telegraph and Telephone Consultative Committee. Information Technology - Open Systems Interconnection - Systems Management: Log Control Function. Technical Report X.735, International Telecommunications Union, 1992.

[33] The International Telegraph and Telephone Consultative Committee. Information Technology - Open Systems Interconnection - Systems Management: Security Alarm Reporting Function. Technical Report X.736, International Telecommunications Union, 1992.

[34] The International Telegraph and Telephone Consultative Committee. Information Technology - Open Systems Interconnection - Systems Management: Object Management Function. Technical Report X.730, International Telecommunications Union, 1992.

[35] Yechiam Yemini. The OSI Network Management Model. *IEEE Communications Magazine*, 31(5):20–29, May 1993.

[36] The International Telegraph and Telephone Consultative Committee. Reference Model of Open Systems Interconnection For CCITT Applications. Technical Report X.200, International Telecommunications Union, 1988.

[37] The International Telegraph and Telephone Consultative Committee. Information Technology - Open Systems Interconnection - Structure of Management Information: Management Information Model. Technical Report X.720, International Telecommunications Union, 1992.

[38] The International Telegraph and Telephone Consultative Committee. Information Technology - Open Systems Interconnection - Structure of Management Information: Definition of Management Information. Technical Report X.721, International Telecommunications Union, 1992.

[39] The International Telegraph and Telephone Consultative Committee. Information Technology - Open Systems Interconnection - Structure of Management Information: Guidelines for the Definition of Managed Objects. Technical Report X.722, International Telecommunications Union, 1992.

[40] The International Telegraph and Telephone Consultative Committee. Specification of Abstract Syntax Notation One (ASN.1). Technical Report X.208, International Telecommunications Union, 1988.

[41] The International Telegraph and Telephone Consultative Committee. Specification of Basic Encoding Rules for Abstract Syntax Notation One (ASN.1). Technical Report X.209, International Telecommunications Union, 1988.

[42] The International Telegraph and Telephone Consultative Committee. Principles For A Telecommunications Management Network. Technical Report M.3010, International Telecommunications Union, February 1992.

[43] The International Telegraph and Telephone Consultative Committee. TMN Interface Specifications Methodology. Technical Report M.3020, International Telecommunications Union, 1992.

[44] The International Telegraph and Telephone Consultative Committee. TMN Management Functions. Technical Report M.3400, International Telecommunications Union, February 1992.

[45] The International Telegraph and Telephone Consultative Committee. TMN Management Capabilities Presented at the F Interface. Technical Report M.3300, International Telecommunications Union, October 1992.

[46] John Mierop, Stefan Tax, and Ronald Janmaat. Service Interaction in an Object-Oriented Environment. *IEEE Communications Magazine*, 31(8):46–51, August 1993.

[47] Menso Appeldorn, Roberto Kung, and Roberto Saracco. TMN + IN = TINA. *IEEE Communications Magazine*, 31(3):78–85, March 1993.

[48] The International Telegraph and Telephone Consultative Committee. Common Management Information Protocol Specification For CCITT Applications. Technical Report X.711, International Telecommunications Union, 1991.

[49] The International Telegraph and Telephone Consultative Committee. Common Management Information Service Definition For CCITT Applications. Technical Report X.710, International Telecommunications Union, 1991.

[50] The International Telegraph and Telephone Consultative Committee. Generic Network Information Model. Technical Report M.3100, International Telecommunications Union, February 1992.

[51] The International Telegraph and Telephone Consultative Committee. Association Control Service Definition for Open Systems InterConnection For CCITT Applications. Technical Report X.217, International Telecommunications Union, 1988.

[52] Walt Dymek. OSI-Based Network Management. In Datapro Information Services Group, editor, *Integrated Network Management*, Datapro Network Management, pages 101–107. McGraw Hill, Incorporated, Delran NJ, June 1993.

[53] Angelo Bean, Desmond Wood, and W. Fairclough. Specifying Goal-Oriented Network Management Systems. *IEEE Communications Magazine*, 31(5):30–36, May 1993.

[54] Won Kim. Architectural Issues in Object-Oriented Databases. *Journal of Object-Oriented Programming*, pages 13–22, 1992.

[55] Dr. Wing Kai Cheng. Distributed Object Database Management Systems. *Journal of Object-Oriented Programming*, 6(1):69–74, March/April 1993.

[56] A.S. Tannenbaum. *Computer Networks*. Prentice Hall, 1991.

[57] M. Bever, K. Geihs, L. Heuser, M. Muhlhauser, and A. Schill. Distributed Systems, OSF DCE and Beyond. In Alexander Schill, editor, *DCE - The OSF Distributed Computing Environment*, number 731 in Lecture Notes in Computer Science, pages 1–20. Springer-Verlag, 1993.

[58] Herve Martin, Michel Adiba, and Bruno Defude. Consistency Checking in Object Oriented Databases: a Behavioral Approach. In Timothy W. Finin and Charles K. Nicholas, editors, *Information and Knowledge Management*, number 752 in Lecture Notes in Computer Science, pages 54–68. Springer-Verlag, 1992.

[59] Shamim Ahmed, Albert Wong, Duvvuru Sriram, and Robert Logcher. A Comparison of Object-Oriented Database Management Systems for Engineering Applications. Technical report, Massachusetts Institute of Technology, May 1991.

[60] Timothy R. Ayers, Douglas K. Barry, John D. Dolejsi, Jeffrey R. Galarneau and Randal V. Zoeller. Development of ITASCA. *Journal of Object-Oriented Programming*, 4(4):46–49, July/August 1991.

[61] W. Kim. *Introduction to Object-Oriented Databases*. M.I.T. Press, Cambridge, MA., 1990.

[62] Itasca Systems, Inc. *Technical Summary for Release 2.0*, 1992. ITASCA Distributed Object Database Management System.

[63] Reda Alhajj, M. Erol Arkun. Queries in Object-Oriented Database Systems. In Timothy W. Finin and Charles K. Nicholas, editor, *Information and Knowledge Management*, number 752 in Lecture Notes in Computer Science, pages 36–52. Springer-Verlag, 1992.

[64] Douglas K. Barry. ODBMS Feature Checklist. Technical Report 1.0, Itasca Systems Inc., March 1992. Itasca Systems Non-Confidential.

[65] Itasca Systems, Inc. *Tools Manual for Release 2.0*, 1992. ITASCA Distributed Object Database Management System.

[66] Lucia Marchisio, Enrico Ronco, and Roberto Saracco. Modeling the User Interface. *IEEE Communications Magazine*, 31(5):68–74, May 1993.

[67] Mark A. Linton, John M. Vlissides, and Paul R. Calder. Composing User Interfaces With InterViews. Technical report, Stanford University, Center for Advanced Systems, Stanford, California. 94305, 1992.

[68] Mike Ahrens. Key Challenges in Distributed Management of Broadband Transport Networks. *IEEE Journal on Selected in Communications*, 12(6):991–999, August 1994.

[69] A.O. Oshinsanwo. The RACE Open Services Architecture Project. *IBM Sys. J.*, 31(4), December 1992.

[70] F. R. McFadden. Conceptual design of object oriented databases. *Journal of Object-Oriented Programming*, pages 23–26, 1992.

[71] Object Management Group. *Object Management Architecture Guide*, November 1992.