NETWORK MANAGEMENT IN THE ESPRIT PROJECT 2100: MAX
Metropolitan Area Communication System

Carlos B. Westphall

Université Paul Sabatier - IRIT
118, route de Narbonne
31062 Toulouse Cédex - France

Abstract

The purpose of this work is to specify an open management architecture that enables the definition and the location of the management elements and their relationships. This architecture provides the basis on which to develop a network management architecture for management networks of the future, and it is based on existing and emerging standards. The paper gives the network management architecture for MAX(Metropolitan Area Communication System), which is a MAN(Metropolitan Area Network). At first, a brief summary of MAX requirements and general architecture is given. Secondly, we specify MAX network management architecture, based on ISO OSI network management and CCITT's TMN(Telecommunications Management Network). Lastly, we focus our study on some elements of the architecture: management information base specification, configuration and fault management processes development, and management processes and their interworking.

1. INTRODUCTION

The development of telecommunication networks from wholly analogue through the ISDN(Integrated Services Digital Networks) of today to the broadband ISDN of tomorrow, and the explosion of new services being offered or planned, have stressed the problem of efficiently administering, operating and maintaining networks and services. In fact, users feel more and more the necessity of providing tools to improve the quality of services and to maximize the network resource performance.

We present our contribution in order to specify a network management architecture for a metropolitan area network. This work, directed by British Telecom Research Laboratories and ALCATEL TITN ANSWARE, was accomplished in collaboration with other partners as part of the ESPRIT programme, Project 2100: MAX(Metropolitan Area Communication System). For this, we consider principally ISO OSI network management and CCITT's TMN standards. TMN and OSI management address different aspects of network management. OSI management provides the means for communicating management information. TMN provides the means for achieving network management. OSI management can be used in a TMN.

The MAX network management general architecture enables the definition and the correct location of the management elements as well as of their relationships, both at the physical and the functional level. The requirements on this architecture are :

This work was supported in part by the CEC(Commission of the European Communities) within the ESPRIT(European Strategic Programme for Research and development in Information Technologies), Project 2100: MAX(Metropolitan Area Communication System), coordinated by CSELT(Centro Studi e Laboratori Telecomunicazioni S.p.A.), Torino, Italy.

- it must allow intra and inter-network management interfaces,
- it should be independent of the size and the topology of the managed network,
- it must be capable of interworking with other standards management systems,
- it must be an open architecture,
- it must be flexible to support the needs of different types of NAMs(Network Access Modules),
- it must permit the identification of specific instance of the generic architecture for a particular MAN.

2. MAX NETWORK DESCRIPTION

The overall objectives of the project MAX (Metropolitan Area Communication System) is to define, design, develop and test a communication system with the following main features:

- large area coverage,
- working at around 600 Mbits/s,
- using fiberoptic and "gate arrays",
- using fast electronic components,
- using advanced multiprocessor architectures and software architectures.

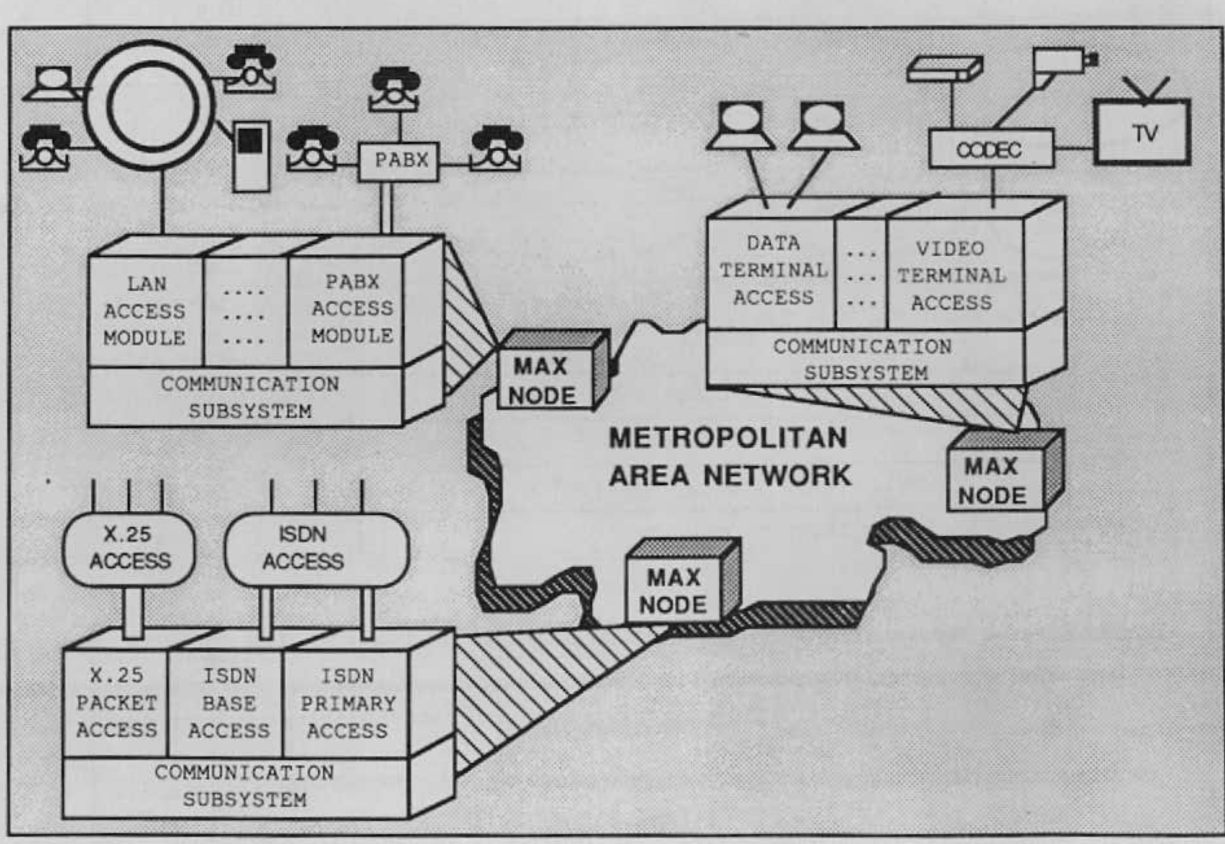

Figure 1. MAX overview.

An overview of the system is depicted in figure 1. This system will provide flexible access to a large number of independent heterogeneous users and cost effective interoperability with public networks.The integration of data, voice, and video services with different needs in terms of bandwidth, error rate, call acceptance, response time, etc... demands for a very flexible information handling technique, able to satisfy the requirements of each service [1].

2.1. MAX node functional architecture

MAX architecture is characterised by the generic node architecture. This architecture will allow each node to receive all the services possibilities and functionalities [1]. The functional architecture of a MAX node is shown in figure 2, in which three macro-levels can be recognized:

- *Level 1:* The lower part of the illustration corresponds to the CS(communication subsystem). The transmission modules perform the functions of the layer 1 of the OSI stack, while the hybrid circuit and packet access modules cover approximately the sublayer 2a [MAC(Medium Access Control)].
- *Level 2:* The communication subsystem provides its services, through the MAC interface, to the NAMs (Network Access Modules). For each type of user subsystem or user device, a specific NAM will be designed and implemented. For example, different NAMs will provide interfaces for: LANs, Voice equipment, data equipment, video codec, ISDN public network. This macro-level represents the "flexibility point" of MAX architecture. All MAX nodes will have the same communication subsystem, but with different NAMs for each interface requirement, allowing a node to be configured.
- *Level 3:* The higher macro-level of figure 2 contains the MAX users. A user can be a simple terminal equipment or a system like a private LAN or a node of a public WAN.

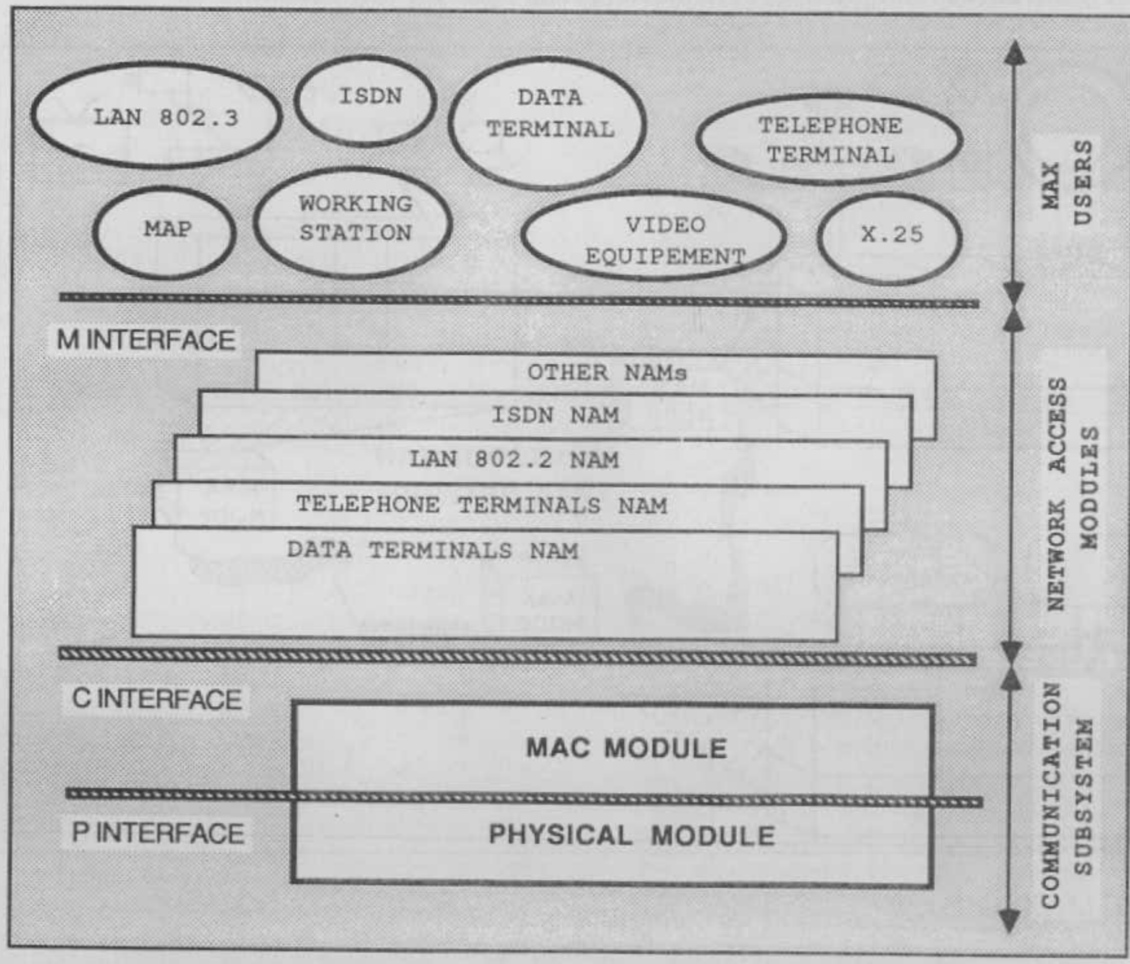

Figure 2. MAX node architecture.

In figure 2, three interfaces delimiting the three macro-levels are identified:

- M-interface, is the boundary between MAX system and the "rest of world". It is not a unique interface, on the contrary, it is only a "reference point" at which the coexistence of a plurality of different physical and logical interfaces is allowed. This will satisfy the main objectives of MAX: every user device has its interface.

- C-interface, is placed between the communication subsystem and the network access modules. Through the C-interface, the following four services are provided to the NAMs:

  . *Isochronous service,* supporting the transparent transfer of data with guaranteed bandwidth and constant end-to-end delay, but without complete error detection capabilities.
  . *Connectionless service,* supporting the transfer of variable length packet data that can tolerate variable end-to-end delivery delay but requiring error detection functions.
  . *Virtual channel service,* supporting the transfer over a virtual channel, of information flows segmented into fixed length cells, having no specified inter-arrival time.
  . *Management and signalling service*, allowing the exchange of both management primitives and signalling primitives (to handle channels and virtual channels) between the communication subsystem CS and the higher part of the node.

- P-interface, is located between the physical layer modules and the MAC layer modules. It performs the functions to the adequate transmission subsystem (these devices could also change during the network evolution, without requiring modifications on the other parts of the nodes).

3. MAX NETWORK MANAGEMENT GENERAL ARCHITECTURE

The MAX network management architecture developed is based upon the principals and ideas of the two most relevant standards; i. e. ISO OSI Management and CCITT's TMN (Telecommunications Management Network). The OSI Management concepts of MIB(management information base), managed objects, the basis of the CMIS, and the ISO 7 layers stack should be adopted [2]. The TMN concepts of network elements, mediation devices, operations systems, Q, X, and F interfaces should also be adopted in the architecture [3]. We can define the general management architecture at three levels:

- physical level, defining the elements, connections and interfaces of the network management system of MAX;
- functional level, defining the MAX network management functional areas and their interactions;
- logical level, defining a logical relationship between some physical or functional elements to define proper logical groupings.

Here, we detail only physical and functional architectures, because the logical architecture includes the two others.

3.1. Physical management architecture

The physical architecture of the MAX network management architecture is developed from the TMN framework as it is shown in figure 3; it can be split into two systems [4]:

a) The LLMS(Lower Layer Management System) formed from the CS(Communication Subsystem), the CSMEs(Communication Subsystem Management Entities) and their communication stacks, the CS-MD(CS Mediation Device), and the part of MAX OS(Operations Systems) that controls the lower layer resources.

b) The HLMS(Higher Layer Management System) which includes the NAMs(Network Access Modules), the NMEs(NAMs Management Entities), the EM(Element Manager), the EMMEs(Element Manager Management Entities), the NAMM-MD(NAM Management Mediation Device), and the part of the MAX OS(Operation System) that controls the higher layer resources.

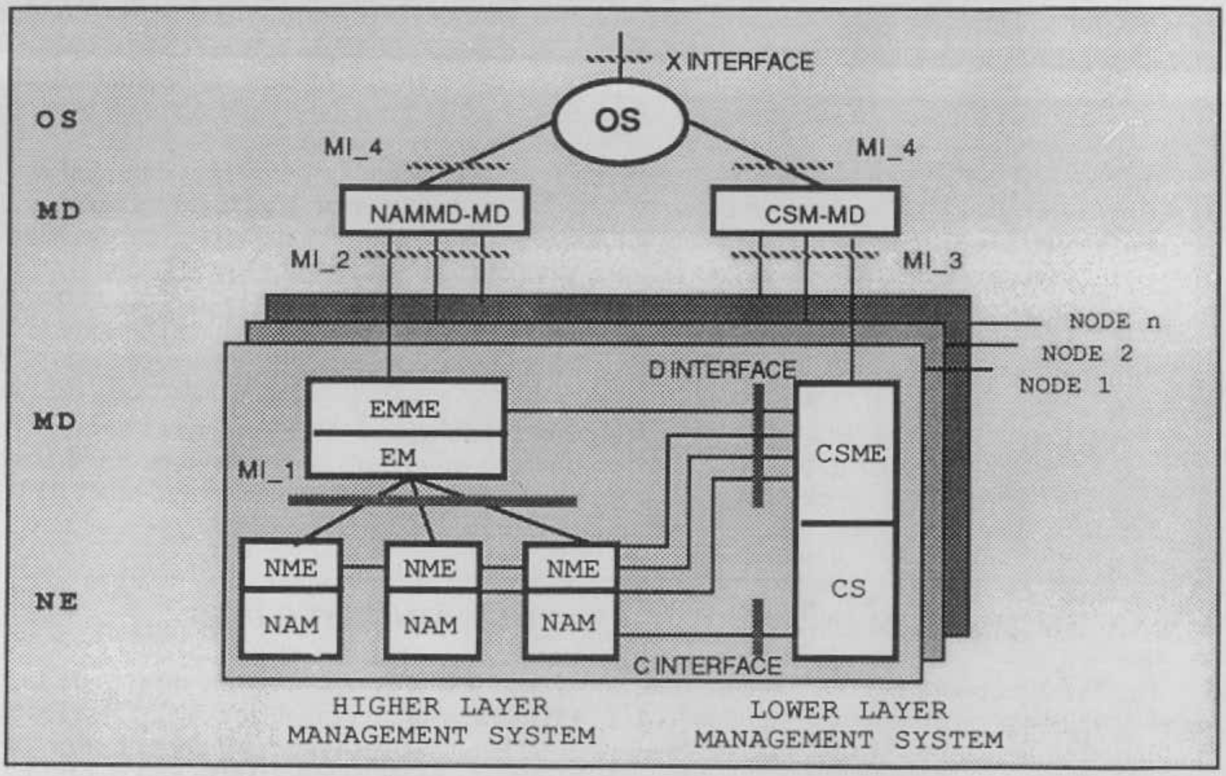

Figure 3. MAX network management physical architecture.

The CS and NAMs are NEs(Network Elements) in TMN terms. The EM(Element Manager) is a simple MD(Mediation Device). The CS-MD and NAMM-MD are more complex mediation devices, implementing some OS type functions. The MAX OS is similar to the TMN OS. MAX elements: the CS, the NAMs and the EM are respectively represented by the management entities CSME, NME, and EMME. Each of this entities controls the resources of the element to manage. Those entities communicate with each other via D interface, while the NAMs communicate with the CS via C interface.

The EM(Element Manager) has an overall view of the resources of the NAMs within the MAX node in which it resides. It is also an agent for the NAMM-MD(NAM Management Mediation Device). The communication between the NMEs and the EM may use the connection less service of the MAC layer of the CS. The MI-1 is an interface between the EMME and the NMEs. The CSMEs are connected to the CSM-MD via the MI-3 interface and the MI-2 is an interface between the NAMM-MD and the EMs. Both CSM-MD and NAMM-MD are connected to the MAX OS via MI-4 interface. The interface to external OSs is the X interface defined in TMN. The MI-x interfaces may be a Q interface. According to the CCITT's Recommendation, protocol stack for this kind of interface is described in Recommendation G.773. We shall use the layer stack showed in figure 4 to exchange information between the MAX OS, NAMM-MD, CSM-MD, NME and CSME.

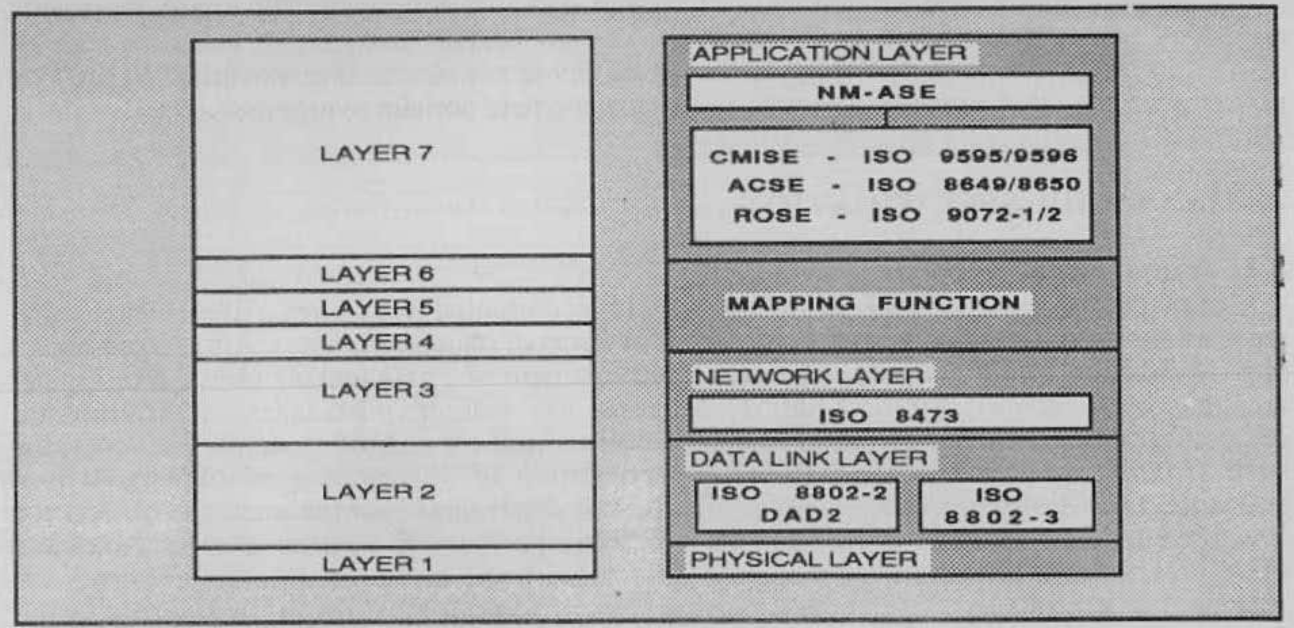

Figure 4. Protocol stacks used for MI-x interface.

3.2. Functional management architecture

For the MAX network management architecture some functional areas have been defined. The specification, relationship and interfaces between these functional areas form the functional architecture [4]:

- CSM(Communication Subsystem Management), is related to the administration of the resources inside the CS module and the CSM-MD;
- NAMM(NAM Management), is related to the administration of all the resources inside the NAMs, the EMs and the NAMM-MD;
- HM(Heterogeneous Management), is related to the control of the resources inside the system allowing the communications between the MAX network management system and external network management systems;
- UEM(User Environment Management), is related to the administration of all the management resources visible to the end user (access control, billing and accounting and special services controlled by the management system);
- GM(Global Management), constituted by all previous management areas, and by the functions allowing the cooperation between areas.

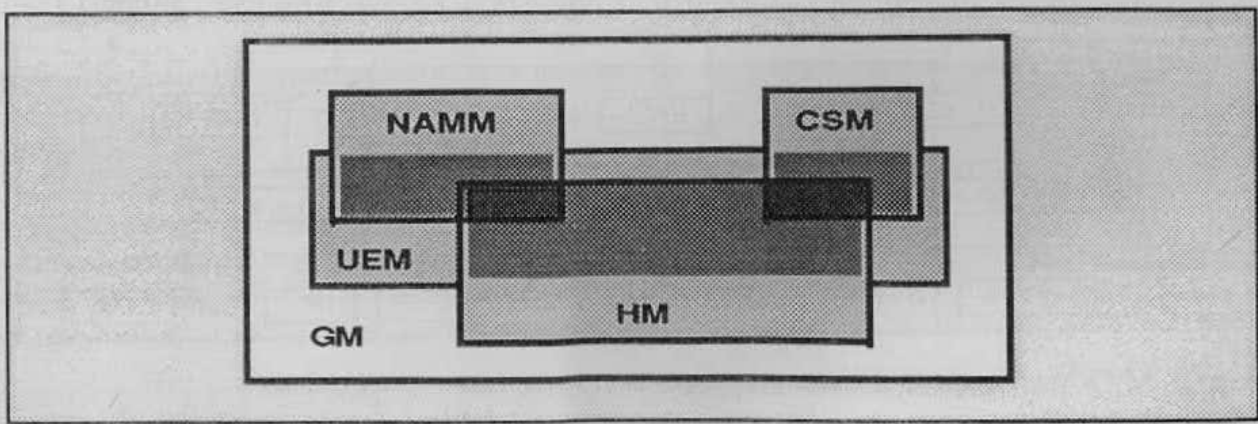

Figure 5. MAX network management functional architecture.

A simple schema for the functional architecture is shown in figure 5. This figure represents the MAX management functional area. It shows how MAX management functions can be divided into sets. The important thing to note is that these management sets overlap, so that one management function can satisfy several MAX management domain requirements.

4. MAX'S MIB SPECIFICATION

4.1. Managed objects

Management information is modelled using object oriented techniques. All elements in the network that are to be managed are represented in terms of managed objects. A managed object is an abstraction for the purposes of network management of a manageable physical or logical resource of the network (Protocol Entities, Modems, Connection...). Management activities are effected trough the manipulation of managed objects in the managed systems. However, the MIB (Management Information Base) is a collection of those managed objects. It is a conceptual repository of management information. It determines how the managed objects are identified and how to define them. Each managed object belongs to a particular class. An object class represents a collection of managed objects with the same, or similar properties. A managed object instance of the class corresponds to some logical or physical entity in the world, or to a relationship between two or more of such entities.

Every managed object maintains a collection of attributes. An attribute consists of a type and one or more values. The values of these attributes at any time constitute some portion of the state and history of the managed object at that time. A number of operations are available that can act on the attributes of managed objects. In addition, a managed object may be capable of performing a range of other actions and reporting events, autonomously informing the outside world of their occurrence. Those events are characteristic of the managed object and are known as notifications. Managed objects participate in relationships with each other. There are two relationships that are of particular importance for management information: the containment relationship and the inheritance relationship.

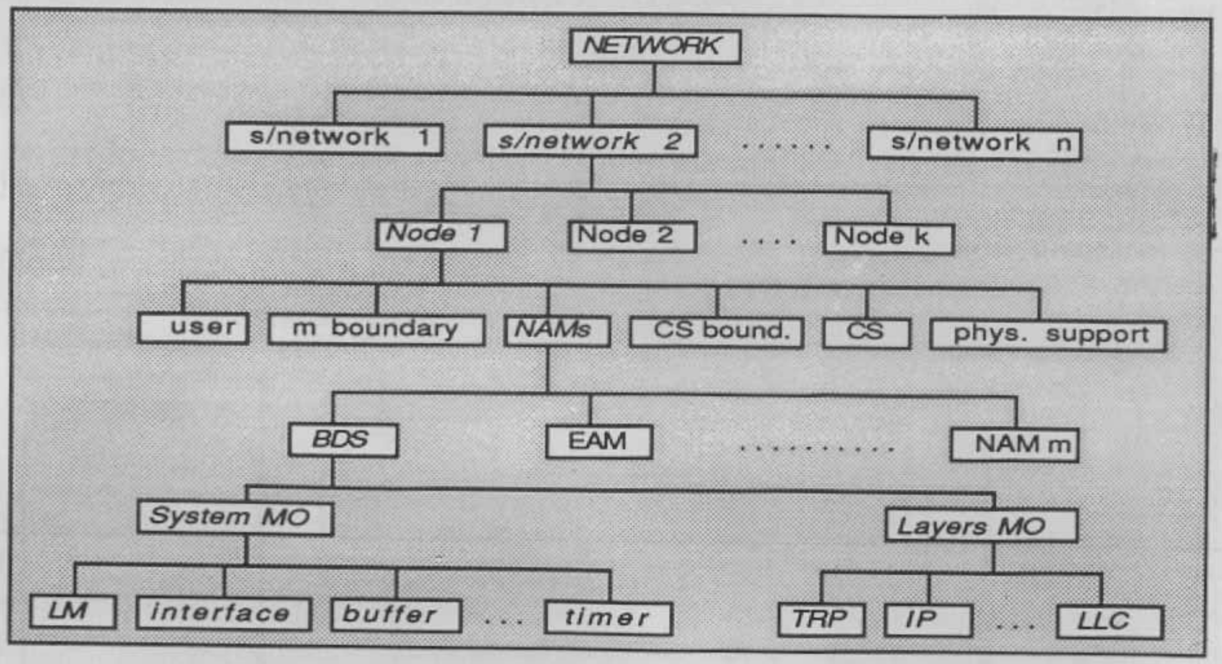

Figure 6. MAX management information tree.

The containment relationship is used to name managed object instances. The hierarchy of superior and subordinate managed object forms a tree, known as the naming tree or the MIT(Management Information Tree), which shows containment relationships between managed object instances. The MAX's MIT structure is depicted in figure 6. The inheritance hierarchy is constructed by applying the relationship "inherits properties of" to object classes. A subclass inherits all characteristics (attributes, operations, behaviour and notifications) of its superclass[5].

4.2. Managed object model

In order to provide a uniform way of writing down the specification of the various categories of information (object classes, operations, etc...), a special notation is provided. The notation used is based upon the ASN.1(Abstract Syntax Notation One)[6,7].

A template is a high-level, informal description that does not include syntactic detail and can be viewed as a form to be filled in by the designer. The formal specification of the objects shall be provided in the form of ASN.1 macros. The MAX managed object template is a generic construct consisting of the following elements: A template type, a list of the key words, and an object identifier. The following general rules should apply to the MAX managed object template:

- If, within completed template, a clause is empty, then the entire clause may be omitted.
- Optional items are placed within square brackets ("[" and "]"). Where an optional item has a default value, that is indicated by underline type ("DEFAULT").
- Where a choice values must or may be made, this is indicated by separating the values with a vertical bar ("value | value").
- In some cases, more than one item can be specified for a clause. Such list within a template are enclosed within braces ("{" and "}"). The items are separated by commas.

In this paper, we shall present only MAX managed object template and its specification in ASN.1. This template is broadly compatible with the procedures and templates under development by ANSI T1M1.5 [8] and ISO WG4 [9].

4.2.1. MAX managed object template

```
MAX MO CLASS
DERIVED FROM { superclass }
SUPERIOR OF { subclasses }
RELATIVE NAME classe-identifier
BEHAVIOUR
  definition
CHARACTERIZED BY
  attributes
  MUST CONTAIN { attribute [qualifier], ... }
  MAY CONTAIN [{ attribute [qualifier], ... }]
OPERATIONS
  [SET TO DEFAULT]
  [GET]
  [ADD]
  [REMOVE]
  [REPLACE]
  [CREATE]
  [DELETE]
  [CONFIRMED-ACTION {action-name,...}]
  [ACTIONS {action-name, ...}]
  [NOTIFICATIONS {notification-name, ...}]
```

4.2.2. MAX macros description in ASN.1.

```
MAX OBJECT-CLASS MACRO ::=
BEGIN
 TYPE-NOTATION ::= subclassof
                   name-bindings
                   relative-name
                   behaviour
                   characteristics
 VALUE-NOTATION ::= value(value-object-identifier)
 subclass-of ::= "DERIVED FROM" value(MAX MO class) | EMPTY
 name-bindings ::= "SUPERIOR OF" value(MAX MO CLASS) | EMPTY
 relative-name ::= identifier
 value-object-identifier ::= STRING
 behaviour ::= "BEHAVIOUR-DEFINITION" value(behaviour-definition)
 behaviour-definition ::= STRING
 characteristics ::= "CHARACTERIZED BY" attribute-list operations
 attribute-list ::= "ATTRIBUTES MUST CONTAIN"
                 "CONFIGURATION ATTRIBUTES" attribute-record
                 "FAULT ATTRIBUTES" attribute-record
                 "SECURITY ATTRIBUTES" attribute-record
                 "PERFORMANCE ATTRIBUTES" attribute-record
                 "ACCOUNTING ATTRIBUTES" attribute-record
 attribute-record ::= mandatory-attribute optional-attribute | EMPTY
 mandatory-attribute ::= "MUST CONTAIN" "{" attributes "}"
 optional-attribute ::= "MAY CONTAIN" "{" attributes "}" | EMPTY
 attributes ::= attribute qualifier
 attribute ::= value(ATTRIBUTE)
 qualifier ::= "READ-ONLY" |"WRITE-ONLY" | EMPTY
 operations ::= "OPERATIONS" create delete get set remove add replace actions notifications
 create ::= "CREATE" | EMPTY
 delete ::= "DELETE" | EMPTY
 get ::= "GET" | EMPTY
 set ::= "SET TO DEFAULT" | EMPTY
 remove ::= "REMOVE" | EMPTY
 add ::= "ADD" | EMPTY
 replace ::= "REPLACE" | EMPTY
 actions ::= "ACTIONS" "{" action-list "}" | EMPTY
 action-list ::= action | confirmed-action | actions
 confirmed-action ::= value(CONFIRMED-ACTION)
 action ::= value(ACTION)
 notifications ::= "NOTIFICATIONS" "{" notification-list "}" | EMPTY
 notification-list ::= notification | notifications
 notification ::= value(NOTIFICATION)
END
```

5. MANAGEMENT PROCESSES DEVELOPMENT

5.1. Configuration and fault management functions

Management application processes may include configuration, fault, performance, accounting, and security. Only configuration and fault are designed and developed. The other functions will be implemented progressively by other partners in the MAX project.

5.1.1. Configuration management

Configuration management is the set of facilities which is used to assist in the management of the normal operations. It includes control of initial values, reset values and default values of managed object attributes, and management of those attributes that govern their normal operations [10]. For MAX, the scope of configuration management facilities is:

- Creation of new instances of a specific object type;
- Suppression of instances of a specific object type;
- Renaming of objects;
- Displaying a list of object instances of a specific type;
- Changing attributes, management state, and relationships of an object;
- Displaying attributes, management state, and relationships of an object;
- Reporting the arrival or removal of an object.

5.1.2. Fault management

Fault management provides support of having managed components (i.e. Managed Objects) send unsolicited error messages to appropriate managing entity, of having the capacity to poll status on a periodic basis and to set error thresholds [11]. For MAX, fault management facilities will encompass:

- Report on spontaneous error;
- Report on threshold alarm exceeding;
- Error historic log (event tracing);
- Repair action reporting facility;
- Resource reinitialization facility;
- Fault management information gathering facility.

5.2. Management processes description

Figure 7 shows the main components of MAX's global and local management. Here, the resources can be the Communication Subsystem or a specific Network Access Module.

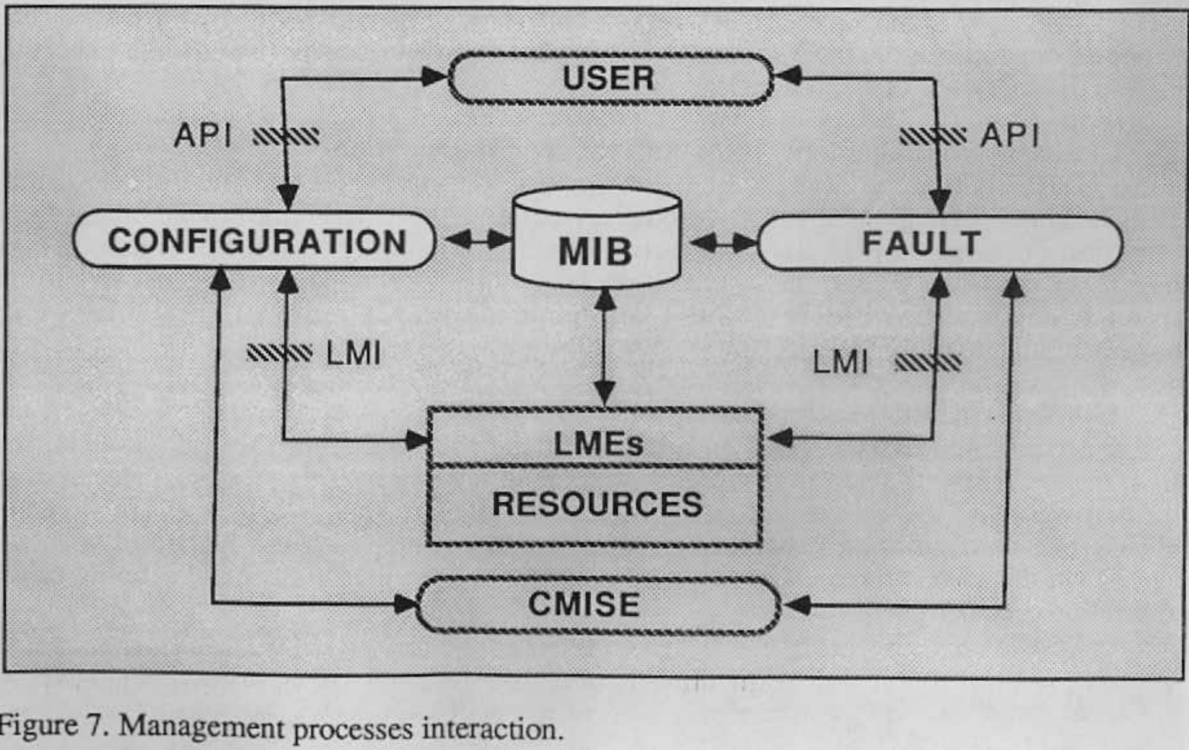

Figure 7. Management processes interaction.

5.2.1. User process

The user process is an interface between the application processes, the operators and/or devices. Its function is to interpret the user commands, to call associate processes, and to send the response to the user. The following algorithm describes the general running of the user process:

```
loop
  wait for a command;  /* from a management user */
  if the command syntax error
     then  send to the  user   the message  "Syntax Error"
     else
        begin
          enable the associate process;
          wait the process response;
          send the response to the user;
        end
end loop.
```

5.2.2. Application process interface

The API(Application Process Interface) is an interface service for the systems management services provided by the management application processes (Configuration and Fault). The API is used to initialize management operations, to receive their results (or responses) and to receive the unsolicited event report.

5.2.3. Layer management entity

The LMEs(Layer Management Entities) provide the specific management functions to the layers where they reside. They provide a local management of layers giving the informations concerning them and provide events control. The services of those entities is accessible to the local entities (Configuration and Fault processes, for example).

5.2.4. Layer management interface

The LMI(Layer Management Interface) is an interface service provided by the LMEs. The LMI nature depends on the layer to manage. It is used for management request in a particular layer, and the reception of event notifications.

5.2.5. Common management information services element

The CMISE(Common Management Information Services Element) module describes CMIS(Common Management Information Service) and the CMIP(Common Management Information Protocol). CMIS defines a set of services, including management association, management notification, and management operation, used by management processes to act upon an agent process. CMIP specifies an application-layer protocol for exchange of management information[12,13].

5.2.6. Configuration process

This process handles the configuration command received from the user process, the CMISE or the LMEs. It also sends them the command responses. This process undertakes to do the hardware and software resources configuration. The information resources are stored in the MIB. The configuration process, therefore communicates with the MIB and with the resources via the LME process. The following algorithm describes the general running of the configuration process:

```
loop
   wait for a command;  /* from the CMISE, the user process or the LME */
   /* command handling */
   execute MAX configuration management facility;  /* see section 5.1.1. */
   if configuration management facility is executed
      then  send the positive response  to the caller ;
      else  send the negative response  to the caller ;
end loop.
```

5.2.7. Fault process

The function of this process is to receive all the commands by the user process, the CMISE or the LMEs. It also sends the necessary notifications to the processes concerned. This process undertakes to identify, and when possible, correct the fault system where they occurs: whether in the hardware or software resources. It also communicates with the MIB and the LMEs. The general running of the fault process is described by the following algorithm:

```
loop
   wait for a command;  /* from the CMISE, the user process or the LME */
   /* command handling */
   execute MAX fault management facility;  /* see section 5.1.2. */
   if fault management facility is executed
      then  send the positive response  to the caller ;
      else  send the negative response  to the caller ;
end loop.
```

5.2.8. Managed object initialization

For example, the MAX's objects showed in figure 6 must be initialized: Network, Node, NAM, CS, the M, C and D interfaces, BDS, LM, Interface, Buffer, Timer, TRP, IP, LLC. In this case the configuration and fault process behaviour is showed bellow:

The configuration process behaviour is:
- initialize the configuration process; /* initial state */
- request the MIB to initialize the all configuration objects;
- request the CMISE to initialize it, and to initialize the remote configuration processes;
- send the result to the user.

The fault process behaviour is:
- initialize the fault process; /* initial state */
- request the MIB to initialize the all fault objects;
- request the CMISE to initialize it, and to initialize the remote fault processes;
- send the result to the user.

6. CONCLUSION

The MAX network management general architecture has been specified in collaboration with all project partners. Subsequently, we have specified the general aspects of management information base, and the configuration and fault management processes for any kind of NAM(Network Access Module). This specification was applied for a specific NAM; the BDS(Bursty Data System). Based on our own specifications and with our close cooperation, ALCATEL TITN ANSWARE, who was developing the BDS NAM, has implemented the BDS's MIB, and the configuration and fault management processes which interact with the BDS's MIB. This implementation has been elaborated in a UNIX environment with

SUNVIEW. In this fashion, a process acts the role of the master and has the task of enabling the other processes, and receiving their state and future results if any from them. The CMISE used is a simulation scenario structure in accordance with ISO OSI/NM protocols.

Up to the time of writing, the design and some implementations of a multi-services communication system, coming from business users located within metropolitan areas, has been carried out in MAX. In the next phase the project will focus on prototype development, integration and proof-of-concept laboratory demonstrations.

7. ACKNOWLEDGMENT

I wish to express my sincere gratitude to Professor J. P. Cabanel, for entrusting me with this project. I would also like to thank my collaborators S. Assoul and E. Feroual, the MAX Network Management Team, and particularly J. G. Barnes and P. J. Willis (British Telecom Research Laboratories), G. Roullet, B. French Keogh and D. Sin (ALCATEL TITN ANSWARE), R. Vercelli and L. Gavi (SIRTI), P. Zøylner (NKT Elektronik), S. Angelopoulos (L-CUBE), A. Ciccardi (Centro Studi e Laboratori Telecomunicazioni) and P. F. Baker (Hewlett-Packard Laboratories).

8. REFERENCES

[1] ESPRIT PROJECT N. 2100: MAX - Metropolitan Area Communication System, Annex I to the MAX contract technical annex. Feb. , 1989.

[2] ISO/IEC 7498-4: Information Processing System - Open System Interconnection - Basic Reference Model Part 4: Management Framework, 1989.

[3] CCITT Recommendation M.30. Principles for a Telecommunication Management Network, 1990.

[4] ESPRIT PROJECT N. 2100: MAX - Metropolitan Area Communication System, Project Deliverable D2: MAX Management System Functional Specification. Apr. 1990.

[5] ISO/IEC DP 10165-1. Information Processing - Open Systems Interconnection - Management Information Service - Structure of Management Information - Part 1: Management Information Model. Jan., 1990.

[6] ISO 8824, Information Processing Systems - Open Systems Interconnection - Specification of Abstract Syntax Notation One (ASN.1). Mar. 1988.

[7] ISO 8825, Information Processing Systems - Open Systems Interconnection - Specification of basic encoding rules for ASN.1. Mar. 1988.

[8] ANSI T1M1.5/89-014R3 - Modelling Guidelines.

[9] ISO/IEC JTC 1/SC 21 N 4065. Information Processing Systems - Open Systems Interconnection - Management Information Services -] Structure of Management Information - Part 4: Guidelines for the Definition of Managed Objects [for CCITT Applications]. Dec. 1989.

[10] ECMA/TC32-TG4/87/63, European Computer Manufacturers Association - Configuration Management Service Definition. Final Draft. Dec. 1987.

[11] ISO/TC97/SC21/WG4 N 418. Forth working draft for the management information services definition -- Part 3: Fault Management. Aug. 1987.

[12] ISO/IEC DIS 9595/2 Information Processing Systems - Open Systems Interconnection - Management Information Service Definition. Dec. 1988.

[13] ISO/IEC DIS 9596/2 Information Processing Systems - Open Systems Interconnection - Management Information Protocol Specification. Dec. 1988.

PS: A previous version of this paper was accepted in the Second International Conference on Local Communication Systems: LAN and PBX, sponsored by IFIP WG 6.4. Here, it is presented as an invited paper.