35

Handling the Distribution of Information in the TMN

Costas Stathopoulos, David Griffin, Stelios Sartzetakis
Institute of Computer Science,
Foundation for Research and Technology - Hellas (ICS-FORTH),
PO Box 1385, Heraklion, GR 711-10, Crete, Greece.
tel.: +30 (81) 39 16 00, fax: +30 (81) 39 16 01.
e-mail: stathop@ics.forth.gr, david@ics.forth.gr, stelios@ics.forth.gr

Abstract

This paper proposes a solution for mapping managed resources (network elements, networks) to the managed objects representing them. It supports an off-line, dynamic negotiation of Shared Management Knowledge in the TMN. Given a method for globally naming managed resources, managers identify the resource they want to manage as well as the management information they require. The manager's requirements are then mapped to the agents which contain the managed objects. From the global name of the agent, and knowledge about the management information that the agent supports, the manager can construct the global distinguished name of managed objects.

The approach uses the OSI Directory where information about managed resources as well as agents and managers is stored. An architecture is described which provides a means of identifying in a global context which agent contains the required management information. Additionally, the architecture provides the abstraction of a global CMIS and the function of location transparency to communicating management processes to hide their exact physical location in the TMN.

Keywords

TMN, systems management, manager/agent model, shared management knowledge, global naming, directory objects, managed objects, location transparency.

1 INTRODUCTION

The M.3010 Telecommunications Management Network (TMN) recommendation (ITU M.3010) describes a distributed management environment: management information for physically distributed network resources and services provided over a large geographical area is maintained on a large number of distributed agents. These agents interact with a variety of management applications over the TMN. The collection of managers and agents (or, with a single name,

A. S. Sethi et al. (eds.), *Integrated Network Management IV*

management processes) in the TMN interact according to the OSI manager/agent model (ISO/IEC 10040). The management information is kept in the agents and consists of managed objects structured hierarchically (ISO/IEC 10165-1) forming the Management Information Tree (MIT). Network resources (network elements or networks) and services being managed, are represented by the managed objects.

A typical TMN implementation may have hundreds of agents. There are proposals (ISO/IEC 10164-16, Sylor 1993, Tschichholz 1993) for the global naming of managed objects. These proposals assume a priori knowledge of which specific agent contains the managed objects in question. This mapping is straightforward in the case where the agent is running on the same system as the managed resources, but in the TMN the mapping may not be as obvious. The general case in the TMN is a "hierarchical proxy" paradigm where Q Adaptors (QAs), Mediation Devices (MDs), and Operations Systems (OSs) are located in separate systems from the Network Elements (NEs). Additionally, the TMN is involved in managing more abstract resources than simple NEs, for example a management process may be interested in networks, services and lower level management processes.

This paper deals with the functionality needed by the TMN in order to efficiently answer the following basic questions: Given a particular managed resource or service that we want to manage (i.e. perform a particular management operation) which is the agent that contains the managed object(s) needed in our management operation? Given that agent, what is the management information base (MIB) it supports? What is the address where the agent is awaiting for requests?

Actually, each of the above questions corresponds to some Shared Management Knowledge (SMK) interrogation (ISO/IEC 10040, NM/Forum 015). Our approach is to provide a global way for referring to elements of the SMK. In order to do so, we use the OSI Directory to register elements of the SMK (such as the mapping from resources to agents, the presentation addresses of management processes and their supported MIBs). Thus, we can achieve an off-line, dynamic SMK negotiation between the management processes.

This paper describes appropriate Directory schemata for storing information about network resources, agents (including the MIBs they support) and managers in the Directory. As a major part of this work, we propose an architecture based on the OSI manager/agent model and the OSI Directory Service. We show how a global Common Management Information Service (CMIS) can be realized and implemented by using this architecture.

We propose a mechanism for supporting the basic function of location transparency. This is one of the distribution transparencies (ITU X.900) necessary in a distributed environment and refers to a location-independent means of communication between management processes, hence hiding their exact location in the TMN.

The OSI Directory Service standard (ITU X.500) describes a specialized database system which is distributed across a network. The Directory contains information about a large number of objects (e.g. services and processes, network resources, organizations, people). The overall information is distributed over physically separated entities called *Directory Service Agents (DSAs)* and consists of directory objects structured hierarchically forming the Directory Information Tree (DIT). The distribution is transparent to the user through the use of *Directory Service Protocol (DSP)* operations between the DSAs. Each directory-user is represented by a *Directory User Agent (DUA)* which is responsible for retrieving searching and modifying the information in the Directory through the use of *Directory Access Protocol (DAP)* operations. The basic reasons for choosing the Directory as the global SMK repository are:

- It provides a global schema for naming and storing information about objects that are highly distributed. For example, every management process in the world can be registered with a unique name (i.e. its Distinguished Name (DN)).
- It provides powerful mechanisms (e.g. searching within some scope in the DIT using some filter) for transparently (through the use of DSP operations between DSAs) accessing this global information.
- One of the major objectives of the OSI Directory, since it was recommended, was to provide an information repository for OSI application processes. For example, by keeping the locations (i.e. OSI presentation addresses) of the various application entities representing the application processes within the OSI environment.

In the following section we describe a way for globally naming managed objects based on registering the management processes in the DIT while in the third section we propose the enhanced manager/agent model that interfaces with the OSI Directory. Putting it all together, section 4 describes the mapping from resources to managed objects and how our enhanced manager/agent model supports the SMK negotiation between two management processes. Next, we present the abstraction of a global CMIS and a location transparency mechanism. Finally, section 6 gives an overview of the implementation of the mechanisms described in this paper.

2 GLOBAL DN FOR MANAGED OBJECTS

The OSI Directory can be used for globally naming application processes in a distributed environment. Any kind of application process can be represented by a directory object that contains information about the process (provided that this information is relatively static). Thus, any application process acting either in the manager or agent role can be globally named. Bearing

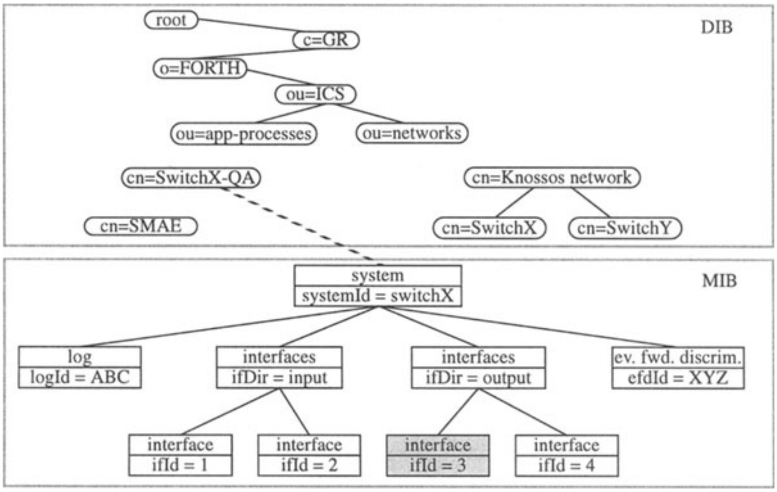

Figure 1 Global DNs for managed objects

in mind that the managed objects use a similar hierarchical naming structure as the directory objects, a common global name space can be realised for both the managed objects and directory objects (Sylor 1993, Tschichholz 1993, and recently ISO/IEC 10164-16).

Figure 1 depicts an example of managed objects named in the global context. Consider the management process that is registered in the Directory Information Base (DIB) with DN:

{C=GR, O=FORTH, OU=ICS, OU=app-processes, CN=SwitchX-QA}

maintaining an MIB containing managed objects that represent some network element (e.g. an ATM switch). Consider a managed object, containing information about interface 3 of the network element, with Local Distinguished Name (LDN) that is, a DN within the scope of the local MIB:

{systemId = SwitchX, ifDir = output, ifId = 3}

This managed object can now be named globally with DN:

{C=GR, O=FORTH, OU=ICS, OU=app-processes, CN=SwitchX-QA, systemId = SwitchX, ifDir = output, ifId = 3}

3 ENHANCING THE MANAGER/AGENT MODEL

In the previous section we described how we can globally name managed objects by exploiting the OSI Directory. In this section we enhance the basic OSI manager/agent model (ISO/IEC 10040) so that a management process can make use of the Directory Service in order to perform systems management functions on the managed objects in the global context.

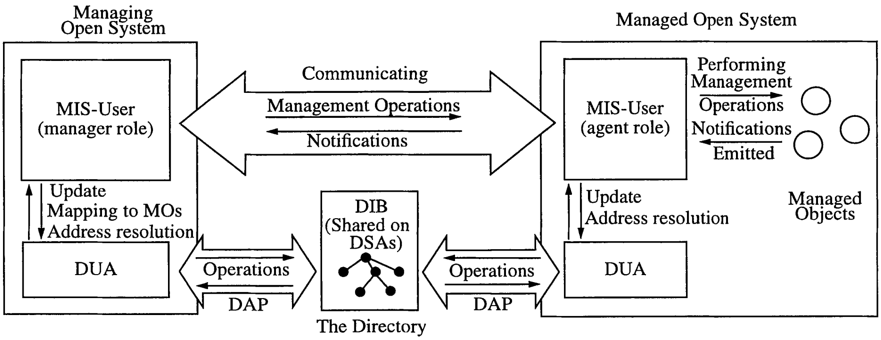

Figure 2 Enhanced manager/agent model

Figure 2 depicts the enhanced manager/agent model. Every open system includes a special purpose DUA. This DUA is responsible for retrieving and updating the information kept in the Directory by issuing DAP operations to the DSAs. In general, the management process uses the DUA for the following:

- Updating the Directory: Management processes should have the capability of updating the Directory by creating, changing or deleting directory objects that represent themselves or other management processes as well as their associated application entities. Although every management process will be able to perform directory updates for its own entry (e.g. on start-up an attribute that marks the process as "running" might be set), it is likely that only special management processes that are responsible for the management of the TMN will fully support this function. These management processes are also responsible for updating the directory objects for the resources with information such as the DN of the management process(es) (acting in the agent role) that represent these resources.

- Mapping to Managed Objects: Every management process acting in the manager role, eventually needs to perform some mapping from the resources it wants to manage to the managed objects (representing the resource) that contain the needed information. This procedure is described in the next section.
- Address Resolution. Every management process, that wishes to make an association with a peer management process, needs a mechanism for finding the presentation address (PSAP) of an application entity representing the latter. Since this address is not always the same for a specific management process, a location transparency mechanism is needed for association establishment. Such a mechanism is described in section 5.

4 MAPPING FROM RESOURCES TO MANAGED OBJECTS

Systems Management deals with management information for physically distributed network resources provided over a large geographical area (divided to many management domains). In general, the relationship between resources and managed objects that represent them is many to many. This means that not only a resource is represented by many managed objects (each one providing a different view of the resource) but also a managed object may represent a collection of resources. Hence, there is no straightforward way for mapping between resources and managed objects that represent them. The knowledge of such a mapping in the TMN is very critical and is actually part of the shared management knowledge because it contains information that must be shared among management processes.

For example, consider the network management case where some decision has to be made about a network reconfiguration due to some network failure. Certain information about the network resources (e.g. network topology information) has to be known in order to discover an optimum reconfiguration solution. This means that having identified the resources that have to be reconfigured, the managed systems that contain the managed objects representing these resources have to be contacted and the appropriate management operations need to be performed. Thus, there must be a way to map from an a priori known resource to some managed object that represent some view of this resource.

In this section we assume a TMN where the management processes communicate based on our enhanced manager/agent model. We describe the information that we have to keep in the Directory for the resources and the management processes and how the latter can use it for performing the above mentioned mapping. Bearing in mind the global naming of managed objects described in section 2, we are going to provide that mapping in the global context.

4.1 Negotiation of Shared Management Knowledge

The mechanisms described in this section provide support for an off-line dynamic negotiation of a part of the shared management knowledge (SMK). In general, the SMK refers to the common knowledge between the application processes performing systems management. This includes (but is not limited to):

- protocol knowledge (e.g. supported application context)
- knowledge of the supported functions (e.g. which management service is provided and to what extent)
- managed object knowledge (e.g. classes and instances)
- knowledge about the relationships between the functions and the managed objects
- knowledge about the mapping of resources to managed objects

For example, to enable management communication between two management processes, prior knowledge such as the MIBs they support and the management activities they can perform is needed. This information can be obtained from the Directory. We use the term "off-line" negotiation of the SMK because it happens prior to the association establishment. It is also dynamic because it happens at run-time. Every management process can update the Directory with management knowledge information and, thus, dynamically modify the SMK that is available to every process.

4.2 Registering network resources in the Directory

Throughout this paper, we use the term "network resources", or, simply, resources, to denote either network elements (e.g. switches) or groups of interconnected network resources (i.e. networks). Given the above definition, network resources can be thought as a containment hierarchy where we have networks containing other (simpler) networks as well as network elements which are always the leaf nodes of a conceptual containment tree.

The Directory can be used for storing information for network resources by registering a directory object for each resource. The containment hierarchy described above together with the existing Directory structure can provide a naming schema for unambiguously identifying network resources. Currently, there is no standard Directory schema for registering network resources in the Directory, although there is ongoing work to that direction (Mansfield 1993) and it is expected that appropriate schemata will exist in the future.

Figure 1 depicts an example of registering a simple network (with two network elements) in the Directory with DN *{C=GR, O=FORTH, OU=ICS, OU=networks, CN=Knossos Network}.*

4.3 Registering the TMN in the Directory

The Telecommunications Management Network is a, possibly separate, network that interfaces a telecommunications network at several different points in order to exchange information with it for management purposes. The TMN is intended to provide a wide variety of management services through the support of a number of management functions.

The TMN physical architecture is composed of a variety of building blocks: Operations Systems (OSs), Mediation Devices (MDs), Q Adaptors (QAs), Data Communication Networks (DCNs), Network Elements (NEs) and Workstations (WSs). Each one of the above building blocks contains a number of TMN functions. For a detailed description of the TMN building blocks and their functions refer to (ITU M.3010).

According to the (ITU M.3020) the overall network management activity is decomposed in areas called TMN management services. The constituent parts of a TMN management service are called TMN management service components. The smallest parts of a TMN management service are the TMN management functions (e.g. performance monitoring).

Additionally, the management functionality may be considered to be partitioned into layers with each layer concerned with some subset of the total management activity. A four-layer management functionality has been identified consisted of the following layers:

- network element management layer, which is concerned with the management of network elements, and supports an abstraction of the functions provided by the network elements,
- network management layer, which is concerned with the management of all the network elements, as presented by the previous layer, both individually and as a set,
- service management layer, which is concerned with how the network level information is utilized to provide a network service, the requirements of a network service and how these requirements are met through the use of the network, and
- business management layer which has responsibility of the total enterprise and is the layer where agreements between operators are made.

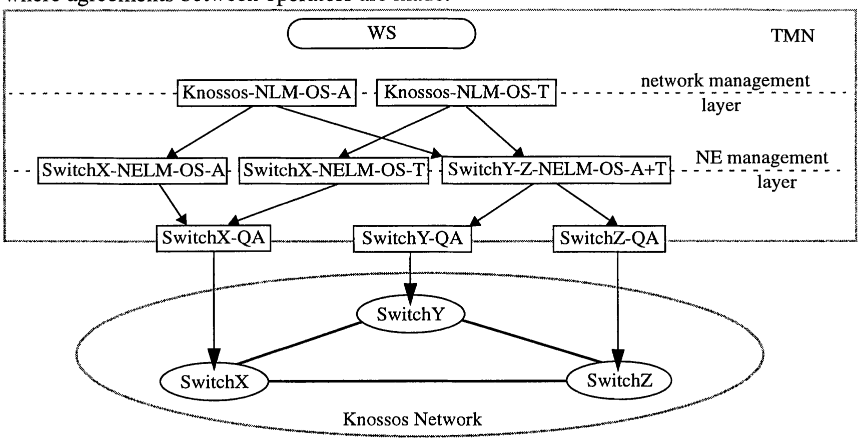

Figure 3 An example of a TMN for a simple network

In the OSI environment we can think of the TMN as a collection of systems management application processes (SMAPs) each one containing one or more Systems Management Application Entities (SMAEs) as defined in (ISO/IEC 10040) in order to accomplish communication between them.

Consider a management domain administered by the organisational unit registered in the Directory with the DN:

{c=GR, O=FORTH, OU=ICS},

a simple network, named "Knossos Network", within the above organizational unit consisted of three switches (NEs) registered in the Directory under the subtree with DN:

{C=GR, O=FORTH, OU=ICS, OU=networks, CN=Knossos Network}

and a TMN in this organizational unit consisted of the following SMAPs (i.e. management processes) (See Figure 3, "An example of a TMN for a simple network"):

- three QAs containing managed objects for the three network elements. (Although a QA may

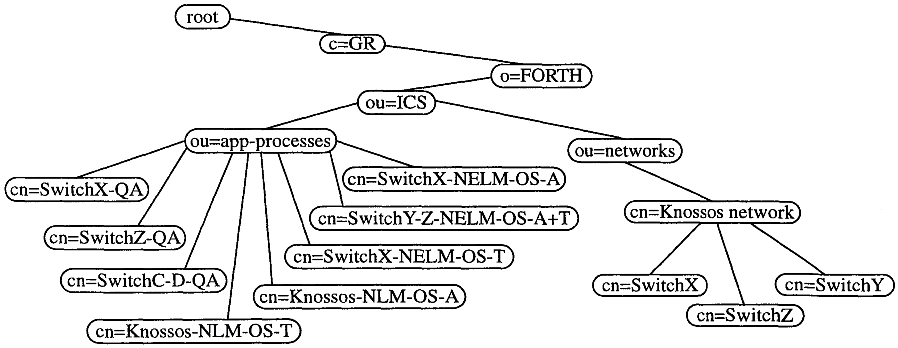

Figure 4 Registering the TMN in the DIB

contain managed objects for more than one network element we only show the simple one-to-one case in this example.),

- three network element level management OSs (NELM-OSs) each one managing a number of network elements in respect to one or two of the *accounting management* and *traffic management* management services. One NELM-OS (namely, the **Switch-Y-Z-NELM-OS-A+T**) manages the two QAs for SwitchY and SwitchZ for both accounting and traffic management. The other two NELM-OSs manage SwitchX, each one for a different management service (namely, **Switch-X-NELM-OS-A** for accounting management and **Switch-X-NELM-OS-T** for traffic management),
- two network level management OSs (NLM-OSs), each one managing the network in respect to one of the above two management services by connecting to the appropriate NELM-OSs (thus, the **Knossos-NLM-OS-A** is for accounting management while the **Knossos-NLM-OS-T** is for traffic management), and
- a WS that is able to manage the resources by connecting to the appropriate OSs or QAs.

In order to register these SMAPs in the Directory we create entries with each entry containing information for a single SMAP. This assigns a global name to every SMAP. Figure 4 depicts the DIT after registering the entries for our TMN example (the network resources are also shown registered in a hierarchy). SMAPs are organized as children of the "*cn=app-processes*" entry under the ICS' entry. Note that we do not register the processes in a hierarchy since this information is going to be obtained from the management services they provide (which includes the management layer on which the SMAPs operate). Also, note that for every SMAP we have to register the entries that contain information about the SMAEs representing the SMAP. Although not depicted in figure 4, these will be registered below the SMAP they represent.

4.4 The Approach to the Mapping Problem

In the TMN, SMAPs (WSs, OSs, QAs, MDs) could be located in separate systems from the resources they represent. This means that even though we know the resource that we want to manage, this does not give any information about the agent that keeps the managed objects that represent the resource. Additionally, there may be more than one agent providing different management services for a resource. In the mapping problem introduced at the beginning of this section we assume that we initially know a global name (namely, the DN of a directory object) for the resource that we want to manage.

Our basic requirement is to provide to every SMAP, acting in the manager role, a mechanism for identifying in the global context the managed objects representing a given resource. Our approach involves the following two-step procedure:

1. Given the DN of a resource and a description of the requested management information that includes:
    - the management service that we want to perform (this will normally be a TMN management service e.g. traffic management),
    - an MIB-independent description of the managed object(s) (this can be based on some abstract description of the object class and the semantics of every managed object. Mechanisms for describing and discovering management information are currently under standardization (ISO/IEC 10164-16)),

    find out the DN of the SMAP that maintains the requested managed object(s) based on the needed management service and by performing a DAP read operation on the resource's directory entry.
2. Perform a DAP read operation on the SMAP you found in the previous step (in case of more than one match, a choice is made based on the MIB that the matching SMAPs support) and identify the LDN(s) of the requested managed object(s) based on
    - the MIB supported by the SMAP and
    - the MIB-independent description of the managed object(s) we have.

    Form the global DN(s) of the managed object(s) you are interested in by concatenating the LDN(s) with the DN of the SMAP.

In order to perform the above procedure, every directory object that represents a resource (either a network or network element) must have a multi-valued attribute that provides the DN of a SMAP that provides some management service for the resource and also identifies which management service this is. That is, a pair of the form: *(DN of agent, Management Service)*. The name of this attribute is "*responsibleSMAP*" and is multi-valued (i.e. many SMAPs can keep managed objects for a single resource in respect to some management service).

Our approach also requires that the following information is kept in every directory object that represents a SMAP:

- an attribute that provides the MIB that the SMAP supports. The name of this attribute is "*supportedMIB*" and is multi-valued (i.e. many MIBs can be supported on a single SMAP). This attribute is present only on SMAPs that are acting on the agent role.
- an attribute that denotes the TMN building block that the SMAP implements. The name of this attribute is "*TMNBuildingBlock*" and is single-valued.
- an attribute for the management service provided by the SMAP. The name of the attribute is "tMNMS" and is multi-valued (i.e. many management services can be provided from a single SMAP).

The value for the *supportedMIB* attribute is a DN. This is the ideal case where the management information is registered under some well-known part of the DIT. The reader can refer to (Dittrich 1993) which describes an approach for registering management schema information in the

Directory. Also, (ISO/IEC 10164-16) recommends the appropriate directory objects for registering the above information in the Directory.

Every directory object that belongs to the standard *applicationEntity* object class should also have attributes with information about the characteristics of the Common Management Information Service Element (CMISE) and the Systems Management Application Service Element (SMASE) of the SMAE. These attributes are discussed in section 5 and are fully described in (ISO 10164-16).

An appendix at the end of this paper contains the ASN.1 definitions for the new attributes. Note that the list for the TMN management services is definitely not complete but rather a small subset of the existing management services (ITU M.3200). Also, since a management service is composed of management service components which, in turn, perform a number of management functions, a Directory schema can be used for registering the hierarchy of the existing TMN management services in the DIT. Finally, every SMAP belongs to the *managementProcess* object class, a subclass of the standard *applicationProcess* class.

5 OSI SYSTEMS MANAGEMENT IN THE GLOBAL CONTEXT

In the previous section we described how the Directory can be used to identify the agent containing specific management information about specific managed resources and how the information about the MIB that the agent supports can be used in the construction of globally unique DNs of the required managed objects. We now show how an OSI SMAP can use DNs in order to issue management operations and notifications in the global context. Additionally, we describe a mechanism for providing location transparency in the proposed manager/agent model (see Figure 2, "Enhanced manager/agent model") for communicating SMAPs.

5.1 The Global CMIS

The Common Management Information Service (CMIS) definition (ISO/IEC 9595) states that, following association establishment between a manager and an agent, the manager issues management operations (while the agent can issue notifications) within the scope of a specific association using LDNs to identify the required managed objects. We can now provide an interface of a global CMIS where the users of the service simply issue CMIS requests using the DN of the managed objects without dealing with the association establishment procedure. For example, (using a simplified semantic notation) a managing open system can issue:

M-GET(DN, attribute_list [,other parameters])

rather than:

A-ASSOCIATE(PSAP_of_agent, &ASSOCIATION_ID)

M-GET(ASSOCIATION_ID, LDN, attribute_list [,other parameters])

which requires that the presentation address (PSAP) of the agent is already known.

On the other hand, a managed open system (i.e. an agent) that reports some notification to a process acting in the manager's role can send the global DN of the managed object that emits the report rather than the LDN. Figure 5 depicts how two management applications communicate using the interface of the Global CMIS.

The Global CMIS uses the Directory to provide a location transparency function. This not only

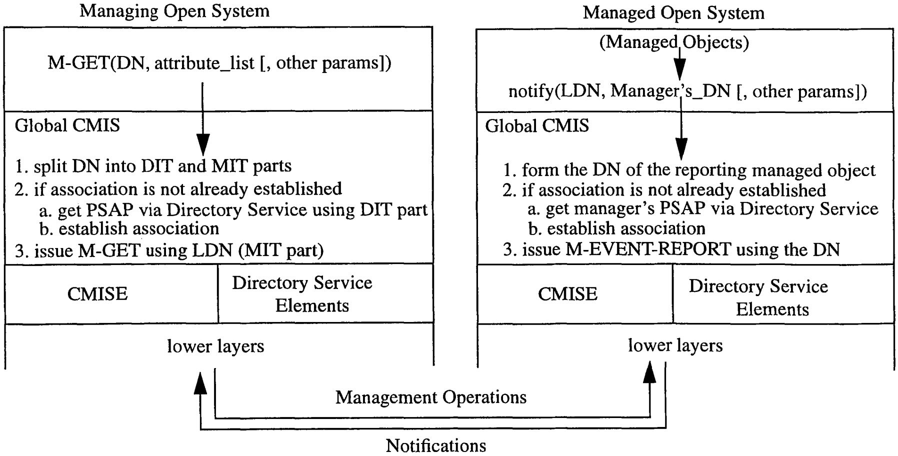

Figure 5 The global CMIS

relieves the management application from the concern of establishing associations with the correct agent but also hides the physical location (PSAP) of the required agents. The management application can assume that managed objects are part of a global and seamless MIB and are identified by their DNs.

5.2 Providing Location Transparency

Location transparency is a basic mechanism in a distributed environment (ITU X.900). In the TMN, it provides a means for finding the address of SMAPs in a location independent way. Bearing in mind that the location of a SMAP may change over time (e.g. a QA for some ATM-switch that is running on machine X might migrate to some other machine if X crashes), we conclude that location transparency should be supported in a TMN. Since the location of a SMAP does not change very frequently, the OSI Directory is appropriate for storing, retrieving and modifying location information for SMAPs.

5.2.1 The Location Transparency Mechanism

The basic requirement for a location transparency mechanism is that, given a SMAP's name, it should provide a means of identifying the location (i.e. OSI presentation address) where the systems management application entity (SMAE) representing that SMAP is awaiting either for management operations or notifications. In the TMN though, there is the possibility that a SMAP is represented from more than one SMAEs. For example, consider the case of a NELM-OS (like the ones depicted in Figure 3) that can act as a manager (by issuing management operations to a QA) and an agent (by serving management requests issued by a NLM-OS) at the same time. Furthermore, there is the possibility that a SMAP supports more than one interoperable interface meaning that a different SMAE might be present for every interface. Additionally, a SMAP that provides some management service can implement a number of management functions. These management functions will be provided by a number of SMAEs representing the SMAP. Bearing these in mind, a location transparency mechanism involves choosing among a number of SMAEs representing the SMAP we wish to communicate with.

In order to provide this functionality, the following information should be kept in every directory object that represents an SMAE:

- the application context supported from the communicating entity. The standard attribute *supportedApplicationContext* will be used for this purpose.
- the presentation address (PSAP) where this SMAE is located. The standard attribute *presentationAddress* will be used for this purpose.

Additionally, every SMAE directory object should contain information regarding the systems management application service element (SMASE) and the common management information service element (CMISE) in the SMAE. The Directory auxiliary object classes *sMASE* and *cMISE* are defined in (ISO 10164-16) for this purpose. They contain attributes that provide information about the supported systems management application service (SMAS) functional units (FUs), the supported management profiles, the supported CMIP version and the supported CMIS FUs on every SMAE.

In our current implementation, every SMAP has the ability to update (either by issuing a DAP modify or DAP add or DAP remove operation) the directory objects that represents itself and its corresponding SMAEs. These update operations take place on start-up or on shut-down of a SMAP. Having the above information about SMAEs registered in the Directory, each SMAP (either in the manager or agent role) can establish an association with a named SMAP after identifying the PSAP of the appropriate SMAE by performing the following (step 2a in figure 5):

1. Given the DN of the SMAP it wishes to associate with, it performs a DAP search under the following conditions:
   - the DN of the SMAP is used as the base object for the search
   - search for objects with the standard application context name "*systems-management*" (defined in ISO 10040)
   - search for objects that support the interoperable interface through which it wishes to communicate (by checking the supported CMIP version and the supported CMIS FUs)
   - search for objects that perform a specific management function in the opposite role (by checking the supported SMAS FUs and the supported management profiles)

   which should return the value of the *presentationAddress* attribute of the matching SMAE.

6 IMPLEMENTATION

The network management platform that is used in the implementation is the OSIMIS platform (Pavlou 1993), developed by the University College of London, which conforms to the CMIP/ CMIS standards (ISO/IEC 9595, ISO/IEC 9596). The Directory Service implementation is based on the ISODE Directory System QUIPU (Kille 1991) version 8.0. A first implementation of the location transparency mechanism has been incorporated into the latest OSIMIS distribution. A full implementation of the mechanisms described in the previous sections is on progress. The performance of the overall system depends heavily on the performance of the QUIPU system which has been analysed and proved satisfactory for our purposes (see also Hong 1993).

7 ACKNOWLEDGMENTS

This work is supported by the CEU RACE project R2059 ICM (Integrated Communications Management). The authors would like to thank all the ICM members for their feedback and support.

8 REFERENCES

Dittrich, A. et al. (1993) Representation of Management Schema Information, TR 6331 GMD-FOKUS.

Hong, W. J. et al. (1993) Integration of the Directory Service in the Network Management Framework, Proceedings of the IFIP TC6/WG6.6 III.

Kille, E. S. (1991) Implementing X.400 and X.500: The PP and QUIPU Systems, Artech House, Boston MA.

Mansfield, G. et al. (1993) Internet working draft: Charting Networks in the Directory.

NM/Forum 015 (1992) Shared Management Knowledge Requirements, OMNIPoint 1.

Pavlou, G. et al. (1993) The OSI Management Information Service User's Manual, Version 1.0.

Sylor, M. (1993) Junction Objects, Proceedings of the IFIP TC6/WG6.6 III.

Tschichholz, M. and Donnelly, W. (1993) The PREPARE Management Information Service, Proceedings of the 5th RACE IS&N Conference.

ISO/IEC CD 10164-16 (1994) Information Technology - Open Systems Interconnection - Systems Management: Management Knowledge Management Function, Denmark.

ISO/IEC 10040 (1992) Information Technology - Open Systems Interconnection - Systems Management Overview, Geneva.

ISO/IEC 10165-1 (1992) Information Technology - Open Systems Interconnection - Structure of Management Information.

ISO/IEC 9595 (1991) Information Technology - Open Systems Interconnection - Common Management Information Service Definition.

ISO/IEC 9596 (1991) Information Technology - Open Systems Interconnection - Common Management Information Protocol Specification.

ITU Recommendation M.3010 (1992) Principles for a Telecommunications Management Network.

ITU Recommendation M.3020 (1992) TMN Interface Specification Methodology.

ITU Recommendation M.3200 (1992) TMN Management Services: Overview.

ITU Recommendation X.900-series (1992) Open Distributed Processing.

ITU Recommendation X.500-series (1988) The Directory.

9 APPENDIX

```
responsibleSMAP ATTRIBUTE
WITH ATTRIBUTE-SYNTAX responsibleSMAPSyntax
MULTI VALUE
```

```
responsibleSMAPSyntax ::= SEQUENCE {
        DistinguishedName, -- DistinguishedName is defined in the standard
        tMNManagementService }
tMNManagementService ::= ENUMERATED {
        Customer Administration (0),
        Management of the security of the TMN (1),
        Traffic Management (2),
        Switching Management (3),
        Accounting Management (4),
        Restoration and Recovery (5) }
managedResource OBJECT-CLASS
        SUBCLASS OF Device -- Device is defined in the standard
        MAY CONTAIN {responsibleSMAP}
supportedMIB ATTRIBUTE
WITH ATTRIBUTE-SYNTAX DistinguishedNameSyntax
MULTI VALUE
tMNMS ATTRIBUTE
WITH ATTRIBUTE-SYNTAX tMNManagementService
MULTI VALUE
tMNBuildingBlock ATTRIBUTE
WITH ATTRIBUTE-SYNTAX tMNBlockSyntax
SINGLE VALUE
TMNBlockSyntax ::= ENUMERATED {
        NE (0), QA (1), MD (2), SL-OS (3), NL-OS (4), NE-OS (5), WS (6) }
managementProcess OBJECT-CLASS
        SUBCLASS OF applicationProcess -- applicationProcess is defined in the standard
        MUST CONTAIN {TMNBuildingBlock}
        MAY CONTAIN {supportedMIB, tMNMS}
```

10 BIOGRAPHY

Costas Stathopoulos received the B.Sc. degree in Computer Science from the University of Crete, Greece in 1992. In 1993 he began the M.Sc. degree at the same university in collaboration with the Advanced Networks, Services and Management Group of the ICS-FORTH, Greece where he also works as a Research Assistant on the CEU RACE II ICM project from 1993. He is involved in the project group for TMN platform extensions, and specifically in providing distribution transparencies and metamanagement support. His main research interests are internetworking, network management, directory services and distributed systems.

David Griffin received the B.Sc. in Electronic Engineering from Loughborough University, UK in 1988. He joined GEC Plessey Telecommunications Ltd., UK as a Systems Design Engineer, where he worked on the CEU RACE I NEMESYS project on Traffic and Quality of Service Management for broadband networks. He was the chairperson of the project technical committee and worked on TMN architectures, ATM traffic experiments and system validation. In 1993 Mr. Griffin joined ICS-FORTH in Crete and is currently employed as a Research Associate on the CEU RACE II ICM project. He is the leader of the project group on TMN architectures, performance management case studies and TMN system design for FDDI, ATM and optical networks.

Stelios Sartzetakis received his B.Sc. degree in Physics and Mathematics from Aristotelian University of Thessaloniki in 1983, and his M.Eng. in Systems and Computer Engineering from Carleton University of Ottawa, Canada in 1986. He worked doing research in communication protocols in Canada. He joined ICS-FORTH in 1988. Today he is research scientist in the networks group responsible for CEU RACE projects in ATM broadband telecommunications networks and services management. Mr. Sartzetakis is responsible for FORTH's telecommunications infrastructure at large. He was principal in the creation of FORTHnet, a multiprotocol, multiservice network, the first Internet access provider in Greece. He served as an independent consultant to private companies and public organizations.