Journal of Network and Systems Management, Vol. 8, No. 2, 2000

Integrated Enterprise Networking Management: Case Study in Intelligent Multimedia Message Handling Systems

Sarandis Mitropoulos[1]

The importance of integrated management stems from the vision of merging telecommunications and distributed computing systems management within the same framework, rapid globalization of world enterprises, and the pressing need for networks and services to be more responsive to "end users." This paper proposes an integrated and distributed management platform which is in line with, but not limited to, the integration of DME, OSI, TMN, and ODP principles. Furthermore, it focuses on a specific layer of the platform describing formally (i) the management domain and management policy concepts; (ii) the services which manage and enforce them; (iii) the performance policy enforcement service; and (iv) how to build-up policy hierarchies. In addition, this paper defines an Intelligent Multimedia Message Handling System (IM3HS) and depicts the high applicability of the domain and policy concepts to its management. Finally, an integrated management architecture for the IM3HS, as well as implementation issues of the IM3HS distributed management platform services are presented.

KEY WORDS: Integrated management; distributed computing and management platforms; management domains; management policies; X.400 MHS; intelligent networks; multimedia systems.

1. INTRODUCTION

Enterprise structures are rapidly changing according to technological, economical, and social progress. The evolution of enterprise networks and systems is driven by the need for enterprise globalization, cost reduction, revenue enhancement, effective customer interaction, and efficient information access. From a technological point of view, universal personal communications and value-added services, as well as networked multimedia applications require more flexible

[1] Department of Electrical and Computer Engineering, National Technical University of Athens, Greece. E-mail: *sarandis@softlab.ntua.gr*

1064-7570/00/0600-0267$18.00/0

and integrated management systems. To this end, some considerable work on ISO/OSI (International Standardization Organization/Open Systems Interconnection), OSF/DME (Open Software Foundation/Distributed Management Environment), OMG/CORBA (Object Management Group/Common Object Request Broker Architecture), ISO/ODP (Open Distributed Processing), TMN (Telecommunications Management Network) and TINA (Telecom Information Networking Architecture) has been done. In this paper a number of important issues of integrated and distributed management of large-scale networked systems are investigated. First, an integrated and distributed management architecture, the domain and policy concepts, as well as their realization are presented. Second, the architecture and the management of an intelligent multimedia electronic mail service are specified and discussed.

2. MANAGEMENT REQUIREMENTS

To have a better understanding of the management requirements of large-scale enterprise networked systems the following main characteristics and needs of these systems must be taken into consideration:

- Existence of numerous resources, like routers, segments, computers, printers, and processes
- Heterogeneity of devices, systems, communication mechanisms and so on
- Distribution of system components, which communicate with each other asynchronously
- Existence of multiple no well co-ordinated management authorities of different technologies
- Need for simple and effective re-engineering process
- Need for reliability, availability, and good performance
- Need for distributed processing transparencies
- Need for guaranteed quality-of-service [1].

These characteristics and needs address the major *management requirements* of well-structured management systems. Specifically, the management systems must be: hierarchical, open and independent from the size and the heterogeneity, integrated from different viewpoints (e.g., functional area, management level), distributed, multiple (many managers can manage concurrently the same object), partitioned, adaptable, reusable, secure, reliable, and of high performance.

Examples of the integration (of DME and OSI) and distribution (in TMN) are given in Fig. 1.

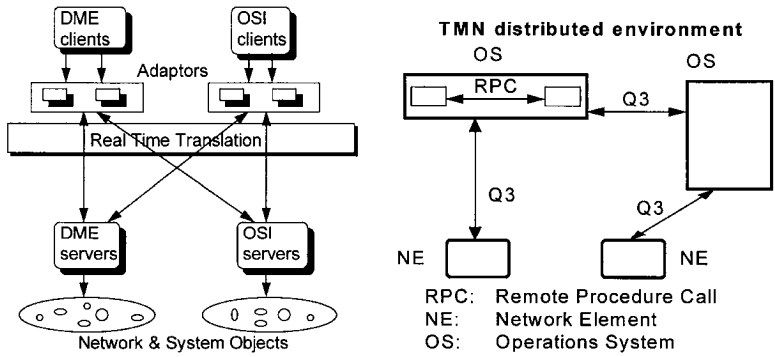


Fig. 1. Integration of DME and OSI, and distribution in TMN.

3. DOMAINS AND POLICIES: STRUCTURING THE MANAGEMENT TASK

Human or automated managers of enterprise networks and systems need tools to structure their management task. Management domain and management policy are two concepts which facilitate this task. A *management domain* (domain) is a collection of managed objects, explicitly grouped together for the purpose of their common management, while a *management policy* (policy) is a set of actions which under constraints affects the behavior of managed objects.

These concepts have been investigated by major standardization bodies (e.g., ISO/IEC JTC/SC21 WG4) and R&D projects (e.g., EU/ESPRIT DOMAINS, IDSM) [2–5].

The following benefits arise by grouping managed objects into domains:

- The borders of policy applicability and management responsibility are explicitly defined
- Management actions are partitioned in different domains
- Policies apply in a set of objects, and not in a single one
- Managers can have different management views
- Domain hierarchies can be created, thus structuring system management hierarchies.

Policies are closely related to the activity of managers. Traditionally, they are coded inside the managers. By separating policies from managers, policies can be treated as objects which can be dynamically changed without re-compilation of managers in the platforms where they are realized. Thus managers could easily be reused in different environments increasing the system portability. Figure 2 depicts the domain and policy concepts as typical management relationships.

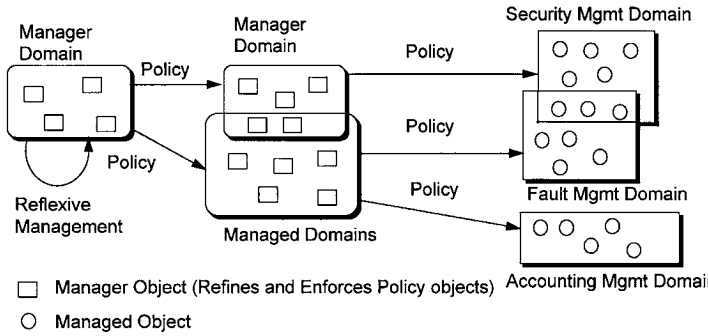


Fig. 2. Domain and policy as management relationships.

4. MANAGEMENT MODELING IN THE ODP ENVIRONMENT

ISO has specified the *System Management Architecture* (SMA) in ISO/OSI 10040 [6]. According to this document, a *managed object* is the OSI management view of a resource within the OSI Environment (OSIE), and is managed through the use of OSI management protocols. The set of managed objects within an open system is called *Management Information Base* (MIB) [7]. A *manager* is a management application which plays a manager role for a specific management interaction. The *Manager Role* includes the ability to invoke operations and to receive notifications. An operation causes the managed object *response* and interactions take place between two parts, that is, a manager and an agent. An *Agent* is a management application which takes the agent role for a specific management interaction. Figure 3 shows the OSI Management Model.

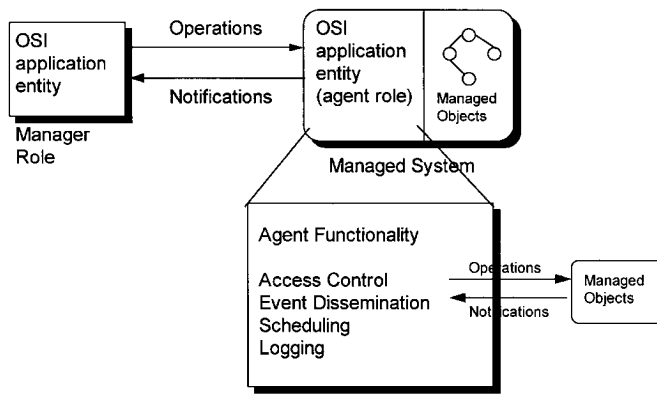


Fig. 3. ISO/OSI management model.

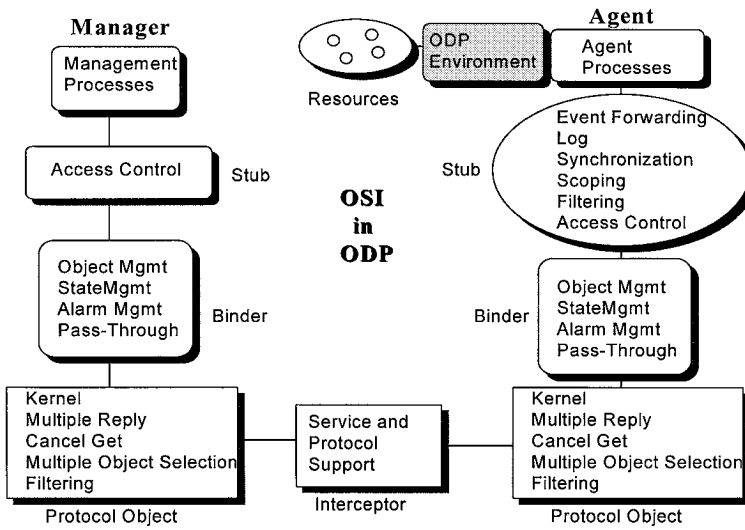


Fig. 4. Realization of OSI management model in ODP environment.

To keep the generality of the Open Distributed Processing Reference Model (ODP-RM) [8], the manager object and the managed object are specified as roles of an object, which are provided through the responsive object interfaces. An object can play different roles from time-to-time or in a specific time. An object can offer manager interfaces, managed object interfaces, and interfaces of the resource functionality. It can also encapsulate other objects. Following the ODP principles to system management, this paper proposes in Fig. 4 an ODP engineering viewpoint of OSI SMA realizing in this way the OSI management concepts in an Open Distributed Management Environment.

Agent, communication management protocol, transparency, binder and stub are engineering concepts that are not parts of the computational viewpoint. From the engineering viewpoint, agent objects can be realized by using stub, binder and protocol objects. Agent functionality, like object creation and deletion, access control, scoping, filtering, synchronization and information dissemination, as well as distributed processing transparencies, must be supported within such objects.

5. INTEGRATED AND DISTRIBUTED MANAGEMENT ARCHITECTURE

A number of standards and efforts are underway to define distributed object frameworks. Some of these standards are specific to management,

whereas other standards are of general purpose. The OSF/DCE (Distributed Computing Environment), OMG/CORBA, ISO/CMIP (Common Management Information Protocol), SNMP (Simple Network Management Protocol), TINA, DEC/EMA (Digital Equipment Corporation/Enterprise Management Architecture), and HP's OpenView are typical examples. *TINA* [9] is a distinguished example which assembles existing thrusts such as TMN and OSI and enhances them through ODP techniques and sophisticated service creation capabilities. Four technology groups are involved in realizing the integration and distribution aspects in TINA: (a) computing and communications (based upon OSI standard); (b) distributed processing and federation (based upon ODP standard; (c) telecom specific application interfaces (based upon OSI and TMN standards); and (d) heterogeneous and dynamic systems modeling (based upon OSI, DTP, OSCA, and others) [10].

To reconcile distributed system environments, there is a need for an object infrastructure, a solid platform with important services and interoperability guarantees, and a comprehensive set of management and object services. Based on these three main aspects, the proposed architecture integrates important technologies, such as DME, OSI, ODP, and TMN, but it is not limited to them.

The proposed integrated and distributed management platform is structured in layers that provide different services. Every layer provides services to the layer above it, or directly to the other upper layers. Figure 5 illustrates the layered structure of the proposed architecture.

A brief description of the architectural layers is provided.

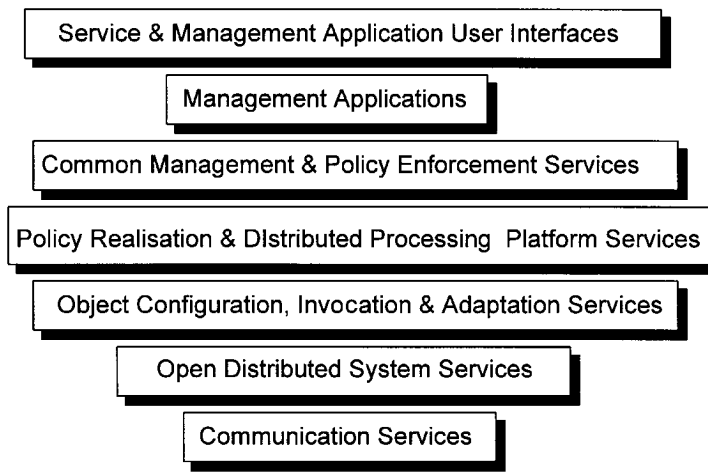


Fig. 5. Layers of the proposed integrated and distributed management architecture.

- *Service and Management Application User Interfaces*: At this layer, a set of interfaces stands offering to the users access to the management applications and services.
- *Management Applications*: According to the five ISO functional management areas, these applications provide configuration, security, performance, accounting, and fault management.
- *Common Management and Policy Enforcement Services*: The services at this layer offer high-level management tools achieving effective and sophisticated management. This layer includes the following services:
  - The *Domain Management* that (a) manipulates domain objects as well as their members, and attributes of a domain object; (b) builds up domain hierarchies; (c) creates/analyses composed domain structures; and (d) maintains persistently the Domain Information Base [11].
  - The *Policy Management Service* that (a) manipulates policy objects and attributes of a policy object; (b) specifies the domains to which a policy applies, as well as which policies apply to a specific domain; and (c) maintains persistently the Policy Information Base [11].
  - The *Active Policy Monitoring Service* that monitors the activity of an active policy object.
  - The *Policy Hierarchy Analysis Service* that analyses the hierarchical relationships between policy objects, and detects possible policy conflicts or inconsistencies.
  - The *Task-Specific Policy Enforcement Services* that depend upon the management task the process (e.g., security), and mainly activate/deactivate task-specific policy objects.

Figure 6 shows the structure of the "common management and policy enforcement services" layer.

- *Policy Realization and Distributed Processing Platform Services*: To enforce a policy, the characteristics of the underlying platform mechanisms have to be known. The platform and its provided mechanisms are object-oriented modeled, and realize policies by performing the appropriate object actions. These underlying mechanisms are distributed and mainly provided by the various processing services of operating systems. Security, time, atomic transaction, directory, file, trading and binding services are typical services provided by this layer, while OSF/DCE [12] and ANSA (Advanced Networked Systems Architecture) are typical distributed processing environments which implement such services.
- *Configuration*, *Invocation*, and *Adaptation Services of Distributed Objects*: At this layer, object invocation services, object class description, object creation and deletion, object starting off and stopping, and generally object configuration services stand. Servers send a request to an

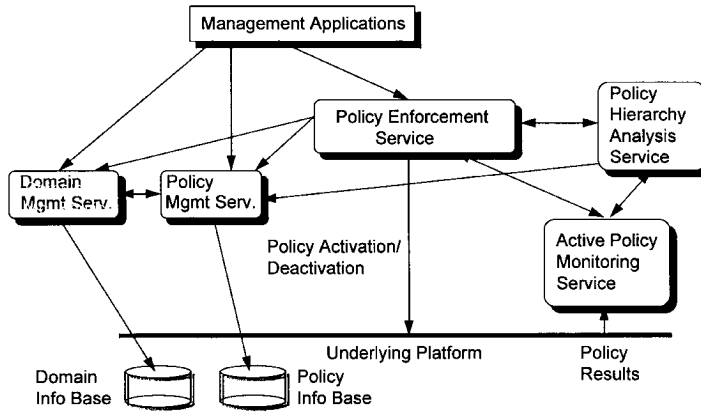


Fig. 6. Interactions at the common management and policy enforcement services layer.

object through an Object Request Broker (ORB) which provides location transparency, and thus the user is now aware of the real object position. Object Adaptors adapt the invocation to the required object type.

This approach is followed by OSF/DME [13]. In addition, OMG/CORBA [14] supports the dynamic invocation, where the object ID passes as a parameter. ISO/OSI supports indirect invocation, because a manager sends CMIP messages to an agent which is responsible for the set of managed objects. Thus, agents are a kind of object servers. CMIP and SNMP support network management object invocations, while XMP (X/Open Management Protocol) offers a common style for CMIP and SNMP access. Figure 7 shows an Object Management Architecture (OMA) for the previously mentioned aspects.

- *Open Distributed System Services*: This layer further refines the open distributed services of the above layer. Specifically, it realizes the location, relocation, migration, access, failure, persistence, replication, and transaction transparencies. In addition, a set of management operations are offered. For example, the following autonomous services or objects are noted: stub objects, binder objects, protocol objects, object creation/deletion, operation requests, access control, scoping, filtering, synchronization, notification dissemination, and dispatching. This layer is a kind of ODP engineering viewpoint of the layer that provides configuration, invocation, and adaptation services of distributed objects and which corresponds to the ODP computational viewpoint.
- *Communication Services*: The lowest level includes communication stacks, such as TCP/IP (Transmission Control Protocol/Internet Protocol) and OSI. Multicast communication mechanisms must also be supported.

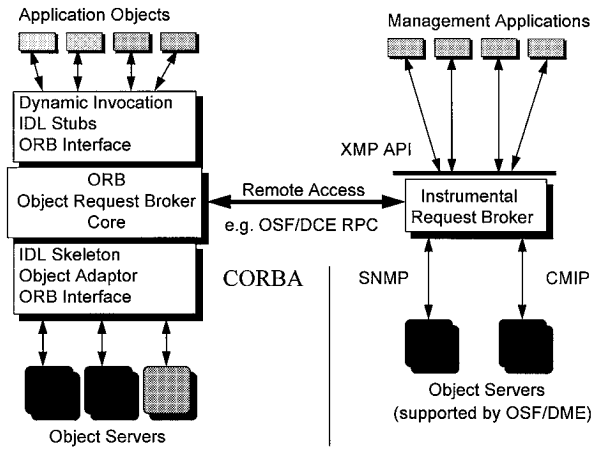


Fig. 7. Object management, invocation and adaptation architecture.

Figure 8 shows a block diagram of the integrated and distributed management architecture of large-scale enterprise networked systems.

6. DESIGNING DISTRIBUTED MANAGEMENT SYSTEMS

The designing process of distributed management systems has many similarities with the designing process of distributed applications, and it concerns management related objects and protocols. To facilitate the designing process of management systems, support tools must be developed to guide the following specifications: management object structures, management object behavior, management data objects, management protocol configuration (specified by data), and management system configuration (logical connections of distributed objects). Figure 9 shows the support of designing process.

7. MANAGEMENT DOMAIN FORMAL SPECIFICATIONS

Formal specifications of the domain object and the domain management service that manipulates domain objects are provided later. A *Domain* is a typical managed object which *maintains a list of references to its member managed objects.* Domain members can be other domains, too. If a domain holds reference to an object, then the object is a *direct member* of that domain and the domain is its *parent.* If a domain is a member of another domain, the first domain is a *subdomain* of its parent (the second one). Members of a subdomain are *indirect* members of the parent domain. Object *ancestors* are all direct and indirect parent domains of the object. If an object is a direct member of multiple domains, the parent domains are said to *overlap* [15]. According to various criteria different

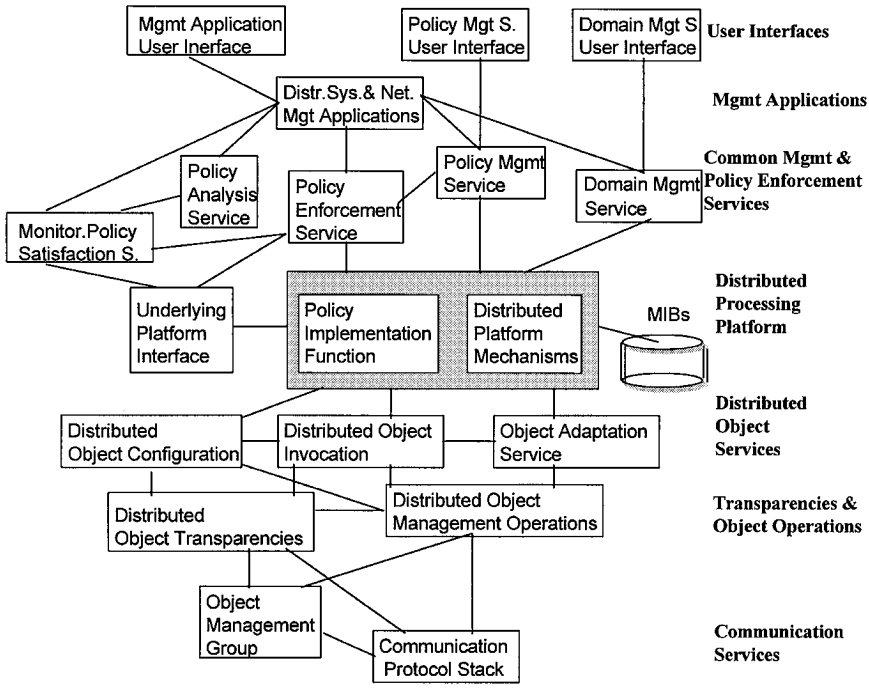


Fig. 8. Integrated and distributed managemnent architecture.

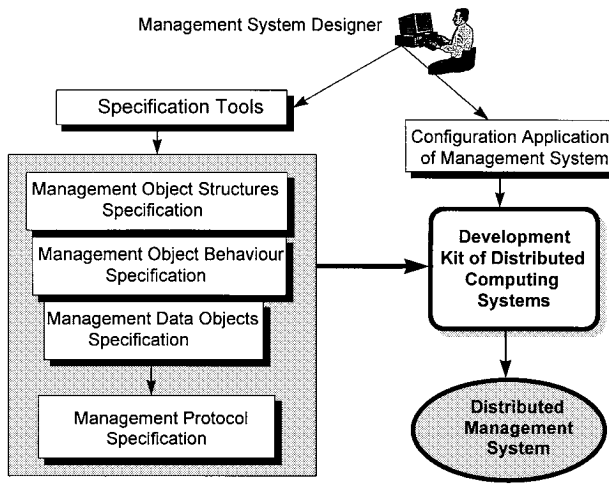


Fig. 9. Architecture for supporting distributed management system design.

kinds of domains can be created such as organizational (e.g., operational structures), geographical (e.g., various sites), informational (e.g., data relationships, flows), computational (e.g., logical connections), engineering (e.g., replications, distribution), technological (e.g., OSI or TCP/IP), as well as domains based on functional areas (e.g., security, accounting) and management levels (e.g., service management level, network management level).

It has been mentioned earlier, that domain objects are manipulated by domain services. Hereafter, (in Scheme I on p. 278) a subset of the GDMO (Guidelines for the Definition of Managed Objects [16]) description regarding the management domain object and the domain management service are provided, followed by their respective containment and inheritance trees.

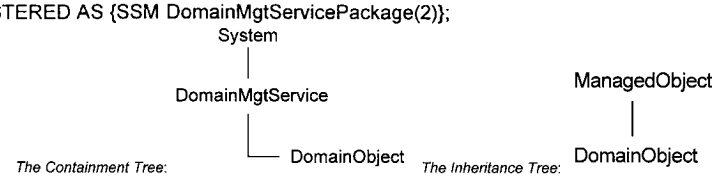
8. MANAGEMENT POLICY FORMAL SPECIFICATIONS

Formal specifications of the policy object and the policy management service that manipulates policy objects are provided later. A *Policy object* is related to at least one domain, and it may define relationships between multiple domains. Policy objects can be grouped into domains, applying in this way policies on policies. A policy can be *propagated* N-levels down through domain hierachies, thus providing an easy way of applying policies. Main attributes of a policy object are the *Subject Manager Set* (enforces the policy), the *Target Managed Set* (shows where the policy has to be enforced), the *Policy Actions* and the *Constrains* under which the actions can be performed. From the *modality* point-of-view, two main categories can be considered, such as the *Obligation Policies* that specify what a manager has to do, and the *Authorization Policies* that specify what a manager is permitted or not permitted to do. From an *Enterprise Hierarchy* point-of-view, five main categories can be taken into account, such as the enterprise goals, the enterprise policies (that try to satisfy the goals), the functional policies (that are related to systems functionality), the policy rules (that are usually automated), and the mechanical information (that realizes in computing systems the policy rules) [17–19]. Hereafter (in Scheme II), a subset of the GDMO description regarding the management policy object and the policy management service are provided followed by their respective containment and inheritance trees.

9. RELATED WORK TO DOMAINS AND POLICIES

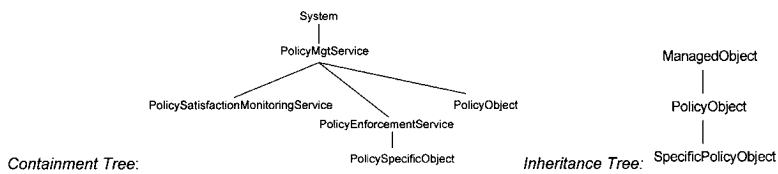
ISO defines a subdomain as a subset. Any policy applying to the superset must apply to the subset. In other words, the definition of subdomains does not permit policy inheritance as an option and does not permit domains to be used to partition responsibility. Furthermore, the domains in ISO terms always include both managers and managed objects. This approach does not permit different policies to apply to managers and the objects they manage.

```
DomainObjectMANAGED OBJECT CLASS
        DERIVED FROM ManagedObject;
        CHARACTERIZED BY
        DomainObjectPackage;
   REGISTERED AS {SSM    DomainMOC(1)};
DomainObjectPackage PACKAGE
   BEHAVIOUR DomainObjectPackageBehaviour BEHAVIOUR
      DEFINED AS
      "Domain Managed Object contains explicitly references to a group of managed objects";;
      ATTRIBUTES
      DomainName              GET-REPLACE
      DomainObjectMember      GET-REPLACE  ADD-REMOVE /*Subdomains*/
      ManagedObjectMember     GET-REPLACE  ADD-REMOVE /* Managed Objects*/
      DomainParentSet         GET-REPLACE  ADD-REMOVE /*Parent Domains*/
      GroupingCriteria        GET-REPLACE  ADD-REMOVE /*Kind of Domain*/
REGISTERED AS {SSM    DomainObjectPackage(1)};
DomainMgtService   MANAGED OBJECT CLASS
        DERIVED FROM ManagedObject;
        CHARACTERIZED BY
           DomainMgtServicePackage;
REGISTERED AS {SSM    DomainMgtServiceMOC(2)};
DomainMgtServicePackage PACKAGE
   BEHAVIOUR DomainMgtServicePackageBehaviour BEHAVIOUR
      DEFINED AS
      "Domain Management Service offers a set of operations on domain objects";;
      ATTRIBUTES
      DomainMgtServiceName     GET-REPLACE
      DomainMgtServiceFederations  GET-REPLACE ADD-REMOVE
      ACTIONS
      CreateDomain,                /* domain object actions */
      RemoveDomain,
      InsertDomainMember,          /*actions on domain members*/
      RemoveDomainMember,
      InsertManagedObjectMember,
      RemoveManagedObjectMember,
      MoveDomainMember,
      MoveManagedObjectMember,
      DecomposeDomainStructure,    /* domain set actions */
      ListDomainMembers,           /*multiple domain member actions*/
      ListManagedObjectMembers,
      ListDescendants-N-levelsDown,
      ....etc.
      SetDomainAttribute,          /* attribute actions */
      GetDomainAttribute,
REGISTERED AS {SSM DomainMgtServicePackage(2)};
```



Scheme I. Part of the GDMO description and the containment and the inheritance trees of management domain object and domain management service.

```
PolicyObject  MANAGED OBJECT CLASS
        DERIVED FROM     ManagedObject;
        CHARACTERIZED BY
            PolcyObjectPackage;
    REGISTERED AS {SSM   PolicyMOC(3)};
PolicyObjectPackage PACKAGE
  BEHAVIOUR PolicyObjectPackageBehaviour BEHAVIOUR
    DEFINED AS
    "Policy Object contains the information which influences the behavior of Target Managed Set";;
    ATTRIBUTES
    PolicyName          GET-REPLACE
    SubjectManagerSet   GET-REPLACE ADD-REMOVE
    TargetManagedSet    GET-REPLACE ADD-REMOVE
    ActionsSet          GET-REPLACE ADD-REMOVE
    ConstrainsSet       GET-REPLACE ADD-REMOVE
    PolicyImportance    GET-REPLACE
REGISTERED AS {SSM   PolicyObjectPackage(3)};
PolicyMgtService MANAGED OBJECT CLASS
        DERIVED FROM     ManagedObject;
        CHARACTERIZED BY
        PolicyMgtServicePackage;
REGISTERED AS {SSM   PolicyMgtServiceMOC(4)};
PolicyMgtServicePackage PACKAGE
  BEHAVIOUR PolicyMgtServicePackageBehaviour BEHAVIOUR
    DEFINED AS
    "Policy Management Service offers a set of operations for policy objects";;
    ATTRIBUTES
    PolicyMgtServiceName          GET-REPLACE,
    PolicyMgtServiceFederations   GET-REPLACE ADD-REMOVE
    ACTIONS
    CreatePolicy,
    RemovePolicy,
    ReadPolicyAttribute,
    WritePolicyAttribute,
    ListDomainsReferredInPolicy,
    ListPoliciesReferredByDomain,
    CheckInternalPolicyConflicts,
REGISTERED AS {SSM PolicyMgtServicePackage(4)};
```



Scheme II. Part of the GDMO description and the containment and the inheritance trees of management policy object and policy management service.

DOMAINS EU ESPRIT project defined an architecture for managing distributed systems. Domains are used to divide the overall management task to sub-tasks. A domain consists of a management part and a set of resources to be managed called Domain's Target. Thus, domains encapsulate manager and

managed objects which make it difficult to support managed objects that are members of multiple domains or to support flexible relationships.

Domino UK project concentrated its research on the policies which are associated with access control rules and on the use of the domain concept to provide a naming context. Domains are a generalization of the tree directory system. Domino project has also investigated the development of an environment with delegation of authority where policies may conflict. For the Domino project, a manager is normally not a member of the domain it manages.

DEC EMA defines a domain as a "sphere of interest of a set of managed objects for a manager." Managers are not part of the domain. Objects are not aware of the domains they belong to. DEC Name Service does not permit cycles and makes use of globally unique names rather than names relative to a domain, and in this manner names are independent of users. In addition, a sub-directory can have only one parent.

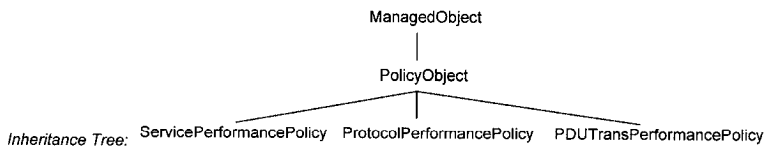
10. PERFORMANCE POLICY ENFORCEMENT SERVICE: A CASE STUDY

Task-specific policy enforcement services provide the appropriate interfaces for the policy activation/deactivation in the underlying platform mechanisms. In this section, the performance policy enforcement service is examined as a case study. The specific policy object is derived from the policy MOC (Managed Object Class). Three network views are considered, that is, the Service, the Protocol and the PDU Transmission views. The performance process of each view includes the monitoring, the analysis and the control to change the network performance. Policy object attributes express QoS indicators related to performance, where their values have to be kept above predefined thresholds. Next (in Scheme III), a subset of the GDMO description of the examined task-specific policy object is provided.

The architecture of the Network Performance Policy Enforcement Service, analyzing mainly the Service Performance Policy Enforcement Service (for brevity called `ServicePerformanceService`), is shown in Fig. 10. In Scheme IV, a subset of the GDMO description of the policy enforcement services related to performance and the respective containment tree are provided.

EFDs (Event Forward Discriminators) operate as event forward agents, forwarding attribute/counter values. At the service level, they mainly forward the values of request rate, service time, error rate, utilization, capacity, and throughput. As far as the connection-oriented protocols are concerned, EFDs mainly forward the values of request rate, utilization, capacity, service time, error rate, and transfer failure. As far as the transmitted data are concerned, they mainly forward the values of PDU (Protocol Data Unit) throughput, utilization in PDUs, capacity in PDUs, error rate and connection establishment delay.

```
ServicePerformancePolicy   MANAGED OBJECT CLASS
          DERIVED FROM      PolicyObject;
          CHARACTERIZED BY
          ServicePerformancePolicyPackage;
   REGISTERED AS {SSM   ServicePerformancePolicyMOC(5)};

ServicePerformancePolicyPackage PACKAGE
   BEHAVIOUR ServicePerformancePolicyPackageBehaviour  BEHAVIOUR DEFINED AS
   "Service Performance Policy Object contains information which influence the performance related
   behavior of the target service";;
      ATTRIBUTES
      ReliabilityQoSGET-REPLACE,
      TransitDelayQoS GET-REPLACE,
      ThroughputQoS  GET-REPLACE;
   REGISTERED AS {SSM   ServicePerformancePolicyPackage(5)};

ProtocolPerformancePolicy  MANAGED OBJECT CLASS
          DERIVED FROM      PolicyObject;
          CHARACTERIZED BY
          ProtocolPerformancePolicyPackage;
   REGISTERED AS {SSM   ProtocolPerformancePolicyMOC(6)};

ProtocolPerformancePolicyPackage PACKAGE
   BEHAVIOUR ProtocolPerformancePolicyPackageBehaviour  BEHAVIOUR DEFINED AS
   "Protocol Performance Policy Object contains information which influences the performance
   related behavior of the protocol machine of the target network";;
      ATTRIBUTES
      TransferFailureQoS  GET-REPLACE,
      ErrorRateQoS     GET-REPLACE;
   REGISTERED AS {SSM  ProtocolPerformancePolicyPackage(6)};

PDUTransPerformancePolicy   MANAGED OBJECT CLASS
          DERIVED FROM      PolicyObject;
          CHARACTERIZED BY
          PDUTransPerformancePolicyPackage;
   REGISTERED AS {SSM  PDUPerformancePolicyMOC(7)};

PDUTransPerformancePolicyPackage PACKAGE
   BEHAVIOUR    PDUPerformancePolicyPackageBehaviour   BEHAVIOUR
   DEFINED AS
   "PDU Transmission Performance Policy Object contains information which influences the PDU    transmission
performance";;
      ATTRIBUTES
      PDULifetimeQoS      GET-REPLACE,
      PDUSizeQoS          GET-REPLACE,
      PDUPriorityQoS      GET-REPLACE;
   REGISTERED AS {SSM  PDUPerformancePolicyPackage(7)};
```



Scheme III. Part of the GDMO description and the inheritance tree of policy objects related to performance.

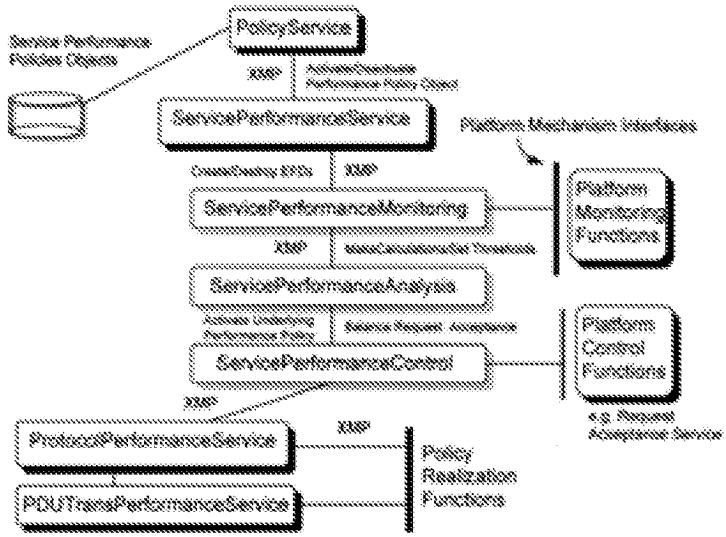


Fig. 10. Performance policy enforcement service.

11. STRUCTURING AND ANALYZING POLICY HIERARCHIES

Policy objects can be structured in hierachies providing a global view of management activity. Modeling objects in hierarchies help to monitor policy satisfaction and policy conflicts. A Policy Hierarchy is defined as:

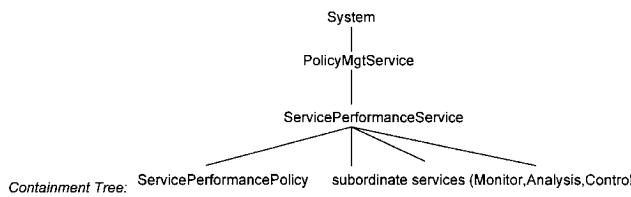
```
PolicyHierarchy::=(Names, Descriptions, Relationship)::=
    (N,D,R),  where:
    N = Set of Policy Object Names (e.g. N={PO1, PO2}),
    D= Policy Object Descriptions (e.g.D={PO1=[Subject_1,
    Target_1, Actions_1, Constraints_1,...],
    PO2=[Subject_2,Target_Actions_2, Constraints_2,...]}),
    R = For each Policy Oject the «meta-policy»set, (e.g.
    R={PO1=[],PO2=[PO1]})
```

A Domain Hierarchy is defined as:

```
DomainHierarchy::=(Names, Descriptions, Relationships)::=
    (N,D,R),
```

Key factors to build up a policy hierarchy are the following: domain hierarchies, manager hierarchies, domain set expressions (e.g., union of domains),

```
ServicePerformanceService MANAGED OBJECT CLASS
        DERIVED FROM      ManagedObject
        CHARACTERIZED BY
            ServicePerformanceServicePackage,
            ServicePerformanceMonitoringPackage,
            ServicePerformanceAnalysisPackage,
            ServicePerformanceControlPackage;
    REGISTERED AS {SSM ServicePerformanceServiceMOC(8)};

ServicePerformanceServicePackage PACKAGE
    BEHAVIOUR  ServicePerformanceServicePackageBehaviour    BEHAVIOUR
    DEFINED AS
     "An object instance in a management platform, offering access to service performance service   through
actions applied on it";;
    ACTIONS
    ActivateServicePerformancePolicy,
    DeactivateServicePerformancePolicy;
    REGISTERED AS {SSM ServicePerformanceSerPackageMOC(8)};

ServicePerformanceMonitoringPackage PACKAGE
    BEHAVIOUR ServicePerformanceMonitoringPackageBehaviour BEHAVIOUR
    DEFINED AS
    "An object instance in a management platform, offering access to monitoring performance service    through
actions applied on it";;
    ACTIONS
    createEFD;
    destroyEFD;
    REGISTERED AS {SSM    ServicePerfMonitoringPackageMOC(9)};

ServicePerformanceAnalysisPackage PACKAGE
    BEHAVIOUR ServicePerformanceAnalysisPackageBehaviour    BEHAVIOUR
    DEFINED AS
     "An object instance in a management platform, offering access to analysis performance service  through
actions applied on it";;
    ACTIONS
    makeCalculations;
    CheckThresholds;
    NOTIFICATIONS
    CalculationsAlarm;
    CalculationsReport;
    REGISTERED AS {SSM   ServicePerfAnalysisPackageMOC(10)};

ServicePerformanceControlPackage PACKAGE
    BEHAVIOUR ServicePerformanceControlPackageBehaviour     BEHAVIOUR
    DEFINED AS
     "An object instance in a management platform, offering access to control performance service    through
actions applied on it";;
    ACTIONS
    ActivateUnderlyingPerformancePolicy,
    BalanceRequestAcceptance,
    REGISTERED AS {SSM    ServicePerfControlPackageMOC(11)};
```



Scheme IV. Part of the GDMO description and the containment tree of policy enforcement services related to performance.

task-specific policies (e.g., performance policies), as well as their interpretation to sub-policies. Taking all these issues into account, an algorithm to build up policy hierarchies is proposed here:

(1) Structure the examined managed system into domains according to various grouping criteria.
(2) Assign manager references to the domains, where a manager with reference to a domain manages it.
(3) Structure the managers of (2) into domains according to various criteria creating an initial manager/managed object hierarchy.
(4) Refine, if necessary, manager domains of (3), using top-down or bottom-up approach and by repeating (1)–(3) steps (here managers are manipulated as managed objects).
(5) Assign a policy to the highest level manager domain.
(6) Interpret the set policy to sub-policies throughout the management hierarchy (including the lowest management hierarchy level) taking into consideration the domain unions, differences and intersections, policy propagation and so on, and thus creating an initial policy hierarchy.
(7) Analyze the initial policy hierarchy in order to improve it. Main policy hierarchy analysis concerns possible policy synergy, policy conflicts, relationships between unrelated policy statements, serialization on policy satisfaction, precedence ordering between competing policies, delegation of responsibility, and so on. For further analysis, see [18].
(8) Refine the policy hierarchy repeating (x)–(7) steps, if necessary.

12. CASE STUDY: INTELLIGENT MULTIMEDIA MESSAGE HANDLING SYSTEMS

Distributed computing applications in modern enterprises use multiple data types like audio, video, moving or still images and so on, and thus require high-quality telecommunication services. In this section, the architecture and the functionality of an Intelligent Multimedia Message Handling System (IM3HS) which handles multimedia messages in an intelligent network environment, are specified.

12.1. X.400 Message Handling System

The X.400 CCITT Recommendations is an International Standard (IS) which concerns the specifications of Message Handling Systems (MHSs) (Fig. 11) [20].

The Message Transfer (MTS) is the core of MHS communication infrastructure. A MTS consists of a set of operational entities and Message Trans-

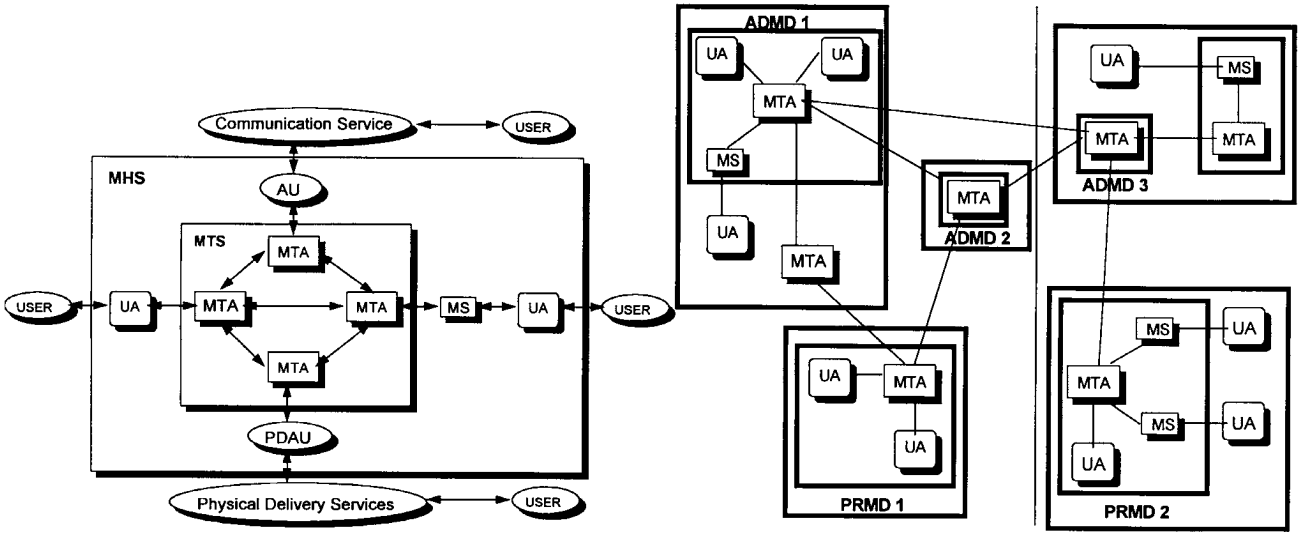


Fig. 11. X.400 message handling system and X.400 administrative domains.

fer Agents (MTAs). The MTAs are interconnected through a telecom network and their functionality is to store-and-forward messages and distribute them to the recipients. System users are connected with MTS by using the User Agent (UA) entities. Each user has its own UA which puts messages in "envelopes" and sends them to the MTA of MTS where they are connected. Receiving messages, UA are responsible for presenting them to users in appropriate forms. Message Stores (MSs) operate like message buffers, so that UAs can retireve messages off-line. Access Units (AUs) convert X.400 messages to forms of other Electronic Message Delivery Systems.

X.400 components are organized in Management Domains (MDs) (Fig. 11). The Administrative Management Domains (ADMDs) are provisioned by PTTs or other X.400 service providers, and are similar to the public mail services. Private Management Domains (PRMDs) are private X.400 components domains of an enterprise. A MD is under the control of a management center which specifies subscribers, addresses, routing algorithms, other interconnected MDs, and so on. Extending the X.400 to transmit multimedia messages, the requirements for bandwidth are increased. A solution to this requirement is to send pure text through MTS, including references (using the Distinguished Object Reference—DOR standard) of the Multimedia parts in a data store (database). User can receive the multimedia information upon request (using the Reference Data Transfer—RDT standard) in a later stage. This approach was followed by the BERKOM European project.

12.2. Intelligent Networks

Intelligent Networks (INs) can be used to configure the MHS according to different e-mail transmission styles, similar to the physical delivery mail services (e.g., normal mail, express mail and so on). In addition, bandwidth on demand has to be according to the transmission style that needs to be supported. Within an Intelligent Network Environment, there are three basic players: the network provider, the service provider and the subscriber [21]. In IN (Fig. 12) consists of 4 interactive parts:

- The Switching Network which contains the Service Switching Points (SSPs).
- The Signaling Network (CSS7) which is a high-capacity and fast network for SSP and IN service connections.
- The Service Control Point (SCP) which offers intelligence and data management.
- The Control System which offers network and service management.

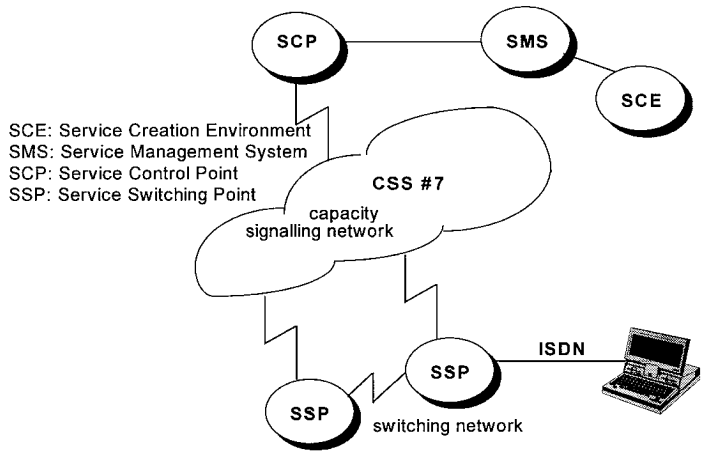


Fig. 12. Intelligent network architecture.

12.3. IM3HS Service Definition and Creation

A major aspect of the IM3HS service provision is the IN Environment which enables the service provider to define, design, develop, test, manage, and maintain the service. A service creation tool is a key tool in developing IN services rapidly and reliably.

From the service definition point-of-view, users interact with the IM3HS service in order to achieve their goals. The telecommunication service characteristics, as well as the interactions with the users, affect the service usability which is not influenced only by the user terminal.

Furthermore, the IM3HS service creation must be based on the concept of the decomposition of service or service features into reusable and network independent building blocks.

Thus, the service will appear in the SCE as a set of graphical icons constituting a decision graph. A visual programming environment in the SCE resembles the global service logic which connects the required service independent building blocks together. The output of the SCE is a decision graph consisting of the flexible service logic and the service support data. The SCE output is finally downloaded to SCP through the Service Management System (SMS) (Fig. 12).

The IM3HS service was defined by the models developed within the framework of the RACE EU Program [22]. Figure 13 shows the architecture of the service.

The Data Bases (DBs) are distributed and interconnected through a WAN. Telecom service provider provides the appropriate bandwidth on demand. Each user reaches the right SCP through SSPs. The SCP servers provide the appro-

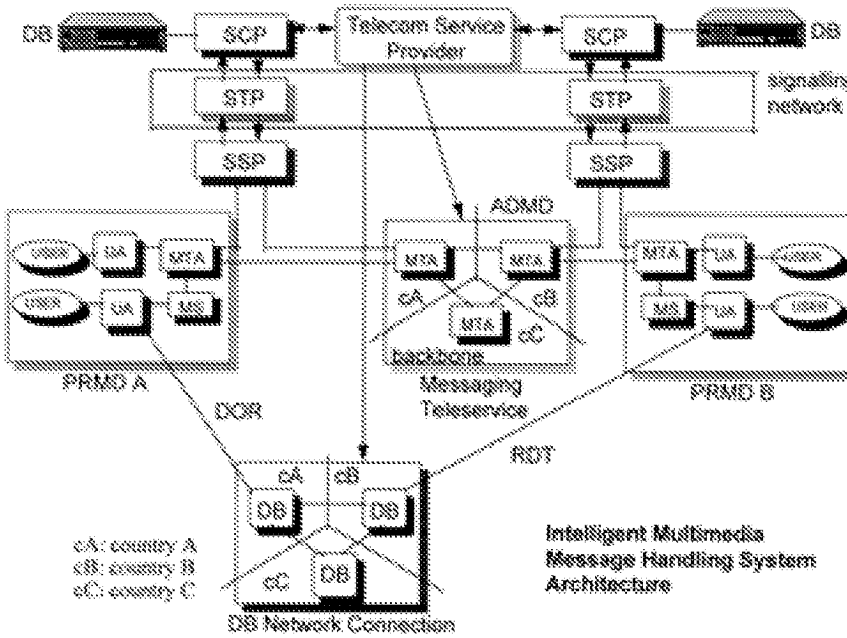


Fig. 13. Intelligent multimedia message handling system architecture.

priate access rights to the subscriber, configure in its service use and perform billing functions.

The service has but it is not restricted to the following tasks as shown in Table I.

12.4. Management of IM3HS

The management of IM3HS must be considered in light of how to deliver excellence in service offerings, concerning not only the actual telecommunications service but also the «whole products»[23]. By selecting an integrated management approach, the control and reliability expected from enterprise-level service management can be delivered [24]. The proposed approach is scaleable, re-usable, and offers flexible delegation of management responsibility.

The systems which have to be managed are the MTS (interconnected MTAs), the Intelligent Network and its services, the Data Bases, the Network (WAN) which interconnects the Data Bases, and the LANs of enterprises which contain multimedia X.400 components. Each subsystem includes resources which are represented as managed objects. These are explicitly grouped together into domains. For instance, in Fig. 14, the X.400 components of an enterprise

Table I. IM3HS Service Tasks

Task description	Type	Information	Acceptable delay	Quality	Volume
send MM mail to a UA	send	multimedia	very much time sensitive	Variable[a]	Variable, 1 MB at least, frequent
send simple mail to UA	send	data, text	time sensitive	data, text	small, very frequent
send MM information to data base	send	multimedia	no too much time sensitive	Variable[a]	Variable, 1 MB at least, frequent
Retrieve MM information from data base	retrieve	multimedia	very much time sensitive	Variable[a]	Variable, 1 MB at least, frequent

[a] Hi-Fi, stand. TV, SVGA.

are grouped into domains, each one of which concerns a different site, while Data Bases are grouped into domains according to different access rules.

12.5. Managed Objects

There are series of known MIBs which formally describe the Managed Objects of IM3HS. These include:

- RFC 1451 Manager-to-Manager MIB, Management System, SNMPv2
- X/Open Management Protocol (XMP) API, components via agents, CMIP/SNMP
- RFC 1565 Network Services Monitoring MIB, Application, SNMPv2
- RFC 1566 Mail Transfer Agent (MTA) MIB, Application, SNMPv2
- RFC 1213 MIB II, Internet Protocol Stack (IPS), SNMPv1
- OSI/NM Forum Library (OMNI-Point1), OSI Protocol Stack, CMIP/AOM12
- RFC 1514 Host Resources MIB, Workstations/Host, SNMPv1
- RFC 1271 Remote LAN Monitoring (RMON), Ethernet LAN/Monitoring, SNMPv1
- RFC 1512 FDDI Interface Type (SMT 7.3), FDDI LAN/Interfaces, SNMPv1

Following are some of the resources modeled as managed objects of each subsystem of IM3HS:

- *MHS*: MTA, UA, MS, MTUser, Message, Association
- *Network*: Switch, Adapter, Router, Link
- *Intelligent Network*: SCP, SSP, CSS7
- *Data Base*: MM DB, IM3HS service DB

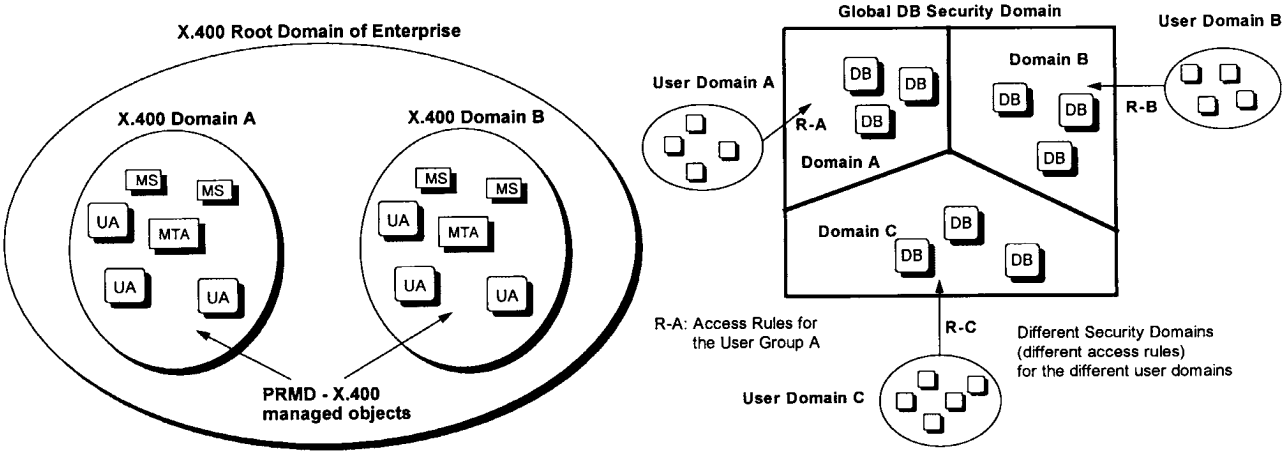


Fig. 14. X.400 domains of an enterprise and DB security domains.

- *Computing System*: Host System
- *Service*: Service, Service Component, Subscriber Service

For example, attributes of the MTA Managed Object Class are the following: message processing capacity, number of delivered messages, number of received messages, number of waiting messages, waiting message volume, Number of MTA Associations, and UA Associations.

12.6. Managers

To create domain-based management hierachies, managers are grouped into domains. A manager domain receives policies from hierarchically higher managers. A scenario for manager domains is described here. At the highest level, there is an integrated manager (IMAN) that is responsible for the management of the whole networked system, and assigns policies to the Network Operator (NetOp), Service Provider (IM3HSP), and Service Subscriber (ServSub). Refining the policies of IMAN, the NetOp applies policies to: (a) the Network Operator which interconnects the data bases (DBNetOp); (b) the IN Manager (InNetMan); and (c) the MTS Manager (MTSMan) (Fig. 15). An example of an obligation policy at this level could be: "The audio/video performance QoS of transmission has to be at the Service Agreement Level (SLA)."

MTSMan receives policies from NetOp, refines them, and sends control commands to a set of network managers. These managers are related to the management functional areas, namely, and Performance (PeMan), Fault (FaMan), Configuration (CoMan), Accounting (AccMan), and Access Control (AcoMan) Managers. Refining PeMan's policies, PeMan applies policies to: (a) PmMan which monitors performance related parameters; (b) PaMan which takes analysis decisions and analyses the monitored results; and (c) PcMan which controls the network performance according to threshold-based criteria (Fig. 16).

This Policy Refinement process continues in a top down approach. For instance, PcMan applies policies to the CallMan which enforces call admission control, the ComMan which controls bandwidth and telecom service provisioning, and the RouMan which enforces routing control. RouMan refining

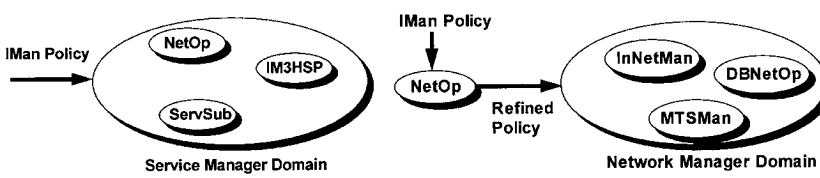


Fig. 15. Service manager domain and network manager domain.

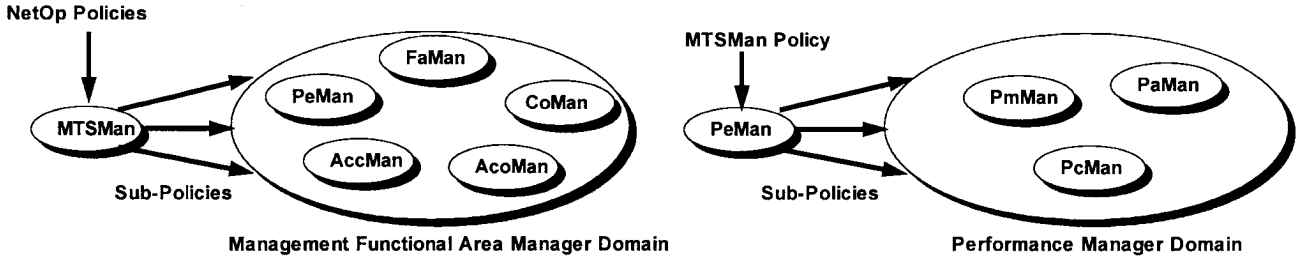


Fig. 16. Management functional area manager domain and performance manager domain.

the PcMan policies produces routing policies, enforcing in such a way routing control to the different technology MTA managers, which consequently affect routing tables of the MTA managed objects (Fig. 17).

Manager domains concerning IM3HS service are shown in Table II.

A list of various managers are described here.

ChargingMan: bills the service subscribers and monitors the whole process.
SupportMan: checks the availability of the IM3HS services.
SubscribeMan: inserts/deletes subscribers and activates users to use IM3HS services.
SSPMan: performs management activites related to SSPs, like SSP configuration, fault detection, correction, and testing and billing of users for SSP use.
SCPMan: maintains operational the SCPs.
MMDBMan: offers normal DBMS functions, checks the integrity, takes checkpoints, performs recovery actions, etc. concerning the multimedia X.400 messages at the data stores. The IM3HSDBMan performs similar functionality to the one of MMDBMan on the IM3HS service data bases.

12.7. The Management Architecture

The proposed architectural framework distinguishes five levels of management similar to TMNs approach. These are the Integrated Management, the Service Management, the Network Management, the Network Element Management, the System/Network Elements (Managed Object with some self-management functionality). Taking into account this approach, as well as the analysis given earlier, Fig. 18 presents an integrated, hierarchical and distributed (e.g., each manager can consist of several distributed manager instances—see that each manager in Fig. 18 consists of several small circles) structure of the management systems (managers) of IM3HS.

The Management Architecture can be realized over a separate physical network. Even if it requires more investment, this approach is beneficial in terms of communication overhead, and robusts against failures in the managed network (e.g., an overlay network which is based on ISDN and it is used for inter-domain management communication independent of the IM3HS system to be managed).

From an end-to-end service management point of view, the aim is to develop a networked management system that can work as an Open Management Platform. A management domain is generally associated with every organization involved in operating subsystems of IM3HS. Its manager is capable of interacting with managers within other management domains at the service management level. The management network formed by the interconnection of the service managers is an end-to-end service management network. The managed objects from each management domain that are visible to service managers

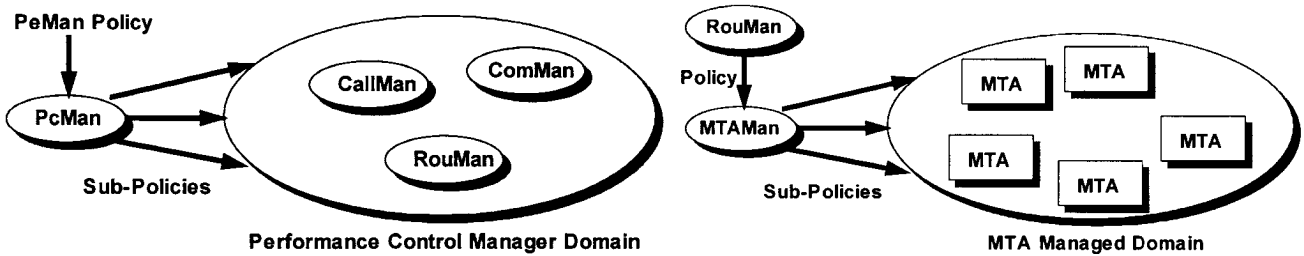


Fig. 17. Performance control manager domain and MTA managed domain.

Table II. Various Manager Domains

Manager domain	Included managers
ISP_Domain	ChargingMan, SupportMan, SubscriberMan, AcoMan, FaMan, . . .
IntelligentNetwork_Domain	SCPMan, SSPMan
Network_Domain	LinkMan, AdaptorMan, SwitchMan, RouterMan
ADMD_Domain	MTAMan, AssociationMan, MTSUserMan, AUMan
PRMD_Domain	UAMan, MSMan, MTAMan, HostSysMan
DB_Domain	MMDBMan, IM3HSDBMan

of other management domains, constitute end-to-end service domains (multiple managers for one domain).

12.8. Implementing Distributed Management Services for the IM3HS

IM3HS management systems use the underlying platform services in order to achieve more flexible and structured management. Distributed management

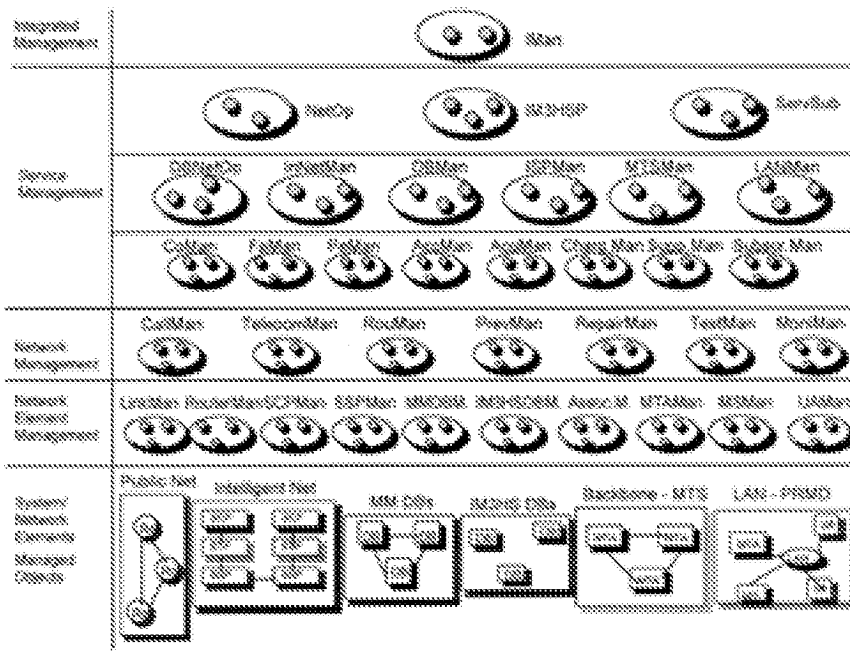


Fig. 18. Integrated and distributed structure of IM3HS management systems.

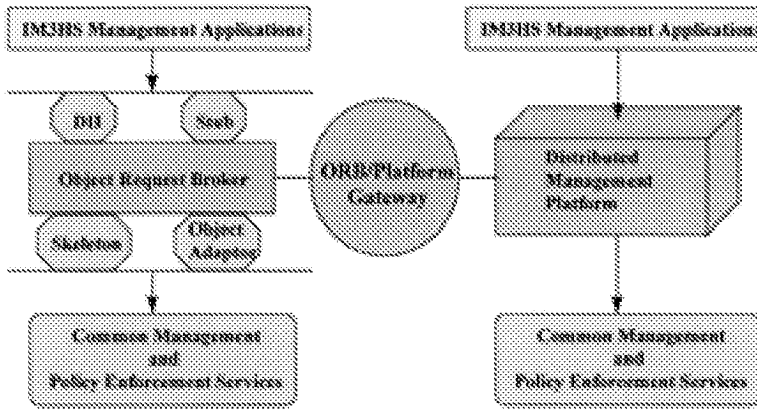


Fig. 19. Architectural issues of distributed management services.

platform services are considered here as the common management and policy enforcement services. These services with the kernel of the open distributed processing environment are available and useable by any other component of the environment.

Considering CORBA as an example, these services can be either integrated into the platform as CORBA objects—if this is the case—or accessed on another external management platform by a CORBA environment through a gateway mechanism. This is an essential requirement because a need may exist for services to be able to communicate with service components which might not be in the same environment. For instance, policy hierarchies can be built over several instances of a distributed policy management service.

A monitoring policy enforcement service (used by PmMan which collects events related to performance parameters) could be one of the management services implemented in CORBA. The role of this service is to activate and de-activate monitoring policy objects. The realization of this service could be based on the creation of EFD-like information from the CORBA agents.

When management service components are constructed as CORBA objects, references must also be also created for these objects, that can have either a standardized format (such as those for the OMG standard Internet Inter-ORB Protocol and the DCE Common Inter-ORB Protocol) or a proprietary format. The operations and the types that a service object supports define the requests that can be made on the object. Both operations and types are described by the object interfaces defined in the OMG Interface Definition Language (OMG IDL).

Figure 19 illustrates the architectural issues of the distributed management services.

The Distributed Management Platform which is communicated with the

CORBA environment through the gateway mechanism, must provide appropriate APIs for interacting with management protocols via industry standards, like X/Open Management Protocol, X/Open Object Management, SNMP and so on. In addition, extensive convenience routines, library modules and applications must be included. Furthermore, the management platform must provide communication stacks, agent/manager services, MIBs and resource access through modules that support specialized access between managed object instance implementations and the resources they represent (e.g., C++ implementations of managed object classes based on GDMO and ASN.1 descriptions) [25].

13. CONCLUSIONS

This article solves a number of problems which arise in the management of large-scale enterprise systems and networks. The proposed platform is based on the Open Distributed Processing principles and on the management domain and management policy concepts. Domains and policies, as well as the services which manage and analyze them were formally specified, and a task-specific policy enforcement service was also defined. Platform services and concepts were applied to the management of an Intelligent Multimedia Message Handling System proving their high applicability.

ACKNOWLEDGMENTS

The author acknowledges the partners Bull, Siemens, IITB, Telesystems, Imperial College, MARI, and PTT NL of the IDSM ESPRIT Project for their contribution in shaping some ideas discussed in this article.

REFERENCES

1. K. Willetts, Service Management: The drive for re-engineering, *Proc. Network Operations and Management Symposium*, IEEE/IFIP, pp. 24–35, 1994.
2. ISO/IEC JTC1/SC21/WG4, Management domain management function, working draft, 1993.
3. ISO/IEC JTC1/SC21/WG4, Third working draft for management domain architecture, 1993.
4. DOMAINS, DOMAINS the principles, Esprit Project 5165, Report, 1992.
5. IDSM, Domain and Policy Service Specification IDSM Deliverable D6, SysMan Deliverable MA2V2, Esprit project 6311, Report, 1993.
6. ISO/IEC JTCI/SC21/WG4, Proposed Draft Amend. to DIS 10040: Systems Management Overview, 1993.
7. ISO/IEC, Information Processing Technology, Open Systems Interconnection, Structure of Management Information: Management Information Model, Report ISO/IEC 10165-1, 1992.
8. ISO/IEC JTC1/SC21/WG7 N885, Basic Reference Model of Open Distributed Processing, Part 1: Overview and Guide to Use, working draft, 1993.
9. *Journal of Network and Systems Management*, Special issue on TINA, Vol. 5, No. 4, December 1997.

10. R. Gupta and P. Cook, Technical assessment of (T)INA-TMN-OSI technology for service management applications, *Proc. Network Operations and Management Symposium*, IEEE/IFIIP, pp. 877–887, 1994.
11. M. Sloman, J. Magee, K. Twidle, and J. Kramer, An architecture for managing distributed systems, *Proc. Fourth IEEE Workshop on Future Trends of Distributed Computing Systems*, Libson, 1993.
12. Open Software Foundation, *Introduction to OSF DCE*, Cambridge Center, 1992.
13. Open Software Foundation, *OSF Distributed Management Environment (DME) Architecture*, 1992.
14. DEC, HP, HyperDesk, NCR, Object-Design, *et al.* The common object request broker: Architecture and specification, Object Management Group, 1991.
15. S. Mitropoulos and W. Veldkamp, Integrated distributed management in interconnected LANs, *Proc. Network Operations and Management Symposium*, IEEE/IFIP, pp. 898–908, 1994.
16. ISO/IEC IS 10165-4: Structure of management Information. Part 4: Guidelines for the definition of managed objects, 1992.
17. M. Masullo, S. Calo, and T. Watson, Policy management: An architecture and approach, *Proc. IEEE Workshop on Systems Management*, UCLA, 1993.
18. J. Moffett and M. Sloman, Policy hierarchies for distributed systems management, *IEEE JSAC*, Special Issue on Management, Vol. 11, 1993.
19. R. Wies, Policies in network and systems management—formal definition and architecture, *Journal of Network and Systems Management*, Vol. 2, No. 1, pp. 63–83, 1994.
20. CCITT, Data Communication Networks—Message Handling Systems, Recommendations X.400-X.420, 1988.
21. H. Fu and C. Jack, An intelligent network service deployment environment, *Proc. Network Operations and Management Symposium*, IEEE, pp. 131–139, 1994.
22. S. Mitropoulos, J.-E. Samouilidis, J. Psaras, and C. Vrouchos, Advanced telecommunications services for multimedia tourism applications, *Proc. MELECON'94*, IEEE, 1994.
23. K. Willetts and E. Adams, Managing a broadband environment: You can't buck the market, *IEEE Communications Magazine*, Vol. 34, No. 12, pp. 108–112, 1996.
24. HP Openview Integrated Network and Systems Management Overview, HP Openview's web site, *http://www.hp.com/openview*, 1997.
25. M. Feridun, L. Heusler, and R. Nielsen, Implementing OSI Agent/Managers for TMN, *IEEE Communications Magazine*, Vol. 34, No. 9, pp. 62–67, 1996.

Sarandis Mitropoulos completed his Ph.D. at the department of Electrical and Computer Engineering of the National Technical University of Athens (NTUA) in 1994. His Ph.D. dissertation focused on Distributed System and Network Management. He received his diploma degree in Informatics and Computer Engineering from the same department of NTUA in 1990. He has been working in technical and project management and business development in European R&TD and integrated solution projects, in the areas of system and network management, and of advanced telecommunications/telematic services, as well as of multimedia technology. He taught at NTUA from 1991 to 1994. He is a member of IEEE, CNOM, and Technical Chamber of Greece.