TOWARDS INTEGRATED NETWORK MANAGEMENT : A DOMAIN/POLICY APPROACH AND ITS APPLICATION TO A HIGH SPEED MULTI-NETWORK

Raouf Boutaba[1] and Simon Znaty[2]

[1] Laboratoire MASI, Université Pierre & Marie Curie,
4, place Jussieu, 75005 Paris, France

[2] Télécom Paris, Département Réseaux,
46, rue Barrault, 75634 Paris cedex 13, France

Abstract

In this paper we introduce a methodology for structuring the management of networked systems. Domains are used as a means for grouping resources according to different criteria. Moreover, we present a management model and the respective activity flows including interactions between managing components. As an example, we apply our concepts to structure the management of a high speed multi-network (ATM, DQDB, FDDI). Emphasis lies on Quality of Service management in the FDDI management domain.

Keywords
Integrated management, policies, domains, multi-network management, QoS.

INTRODUCTION

Networked systems are becoming large, complex and heterogeneous. Moreover they include both communication and information processing. Such a scenario gives rise to several requirements relating to management of these systems. Indeed, to provide their services efficiently, networked systems require management tools to control and maintain in a consistent and a uniform way the large number of resources attached to them. These include network, system and application resources.

The ultimate aim of this work is to ensure the integrated management of all resources in a networked system. Our approach is to define a set of structuring principles that can be used when building the management system. We make a distinction between management policies (i.e. the objectives of management) from the resources and activities being managed (i.e. managed objects). The management system may be structured into management subsystems and the management responsibilities distributed over these subsystems. We allow such a partitioning by use of *domains* as a means for grouping resources for management purposes (e.g. a domain applying a common management policy). The dependencies between the emerging domains allow to model the authority delegation process as well as peer-to-peer interactions between managers. This work is in the scope of the Esprit II project DOMAINS (Distributed Open Management Architecture In Networked Systems). The basic concepts of the DOMAINS project are presented in [1].

In a first section, this paper presents our structuring principles for network management. It focuses on the way management is performed in order to automatise the management process.

In the last section, we apply our concepts to structure the management of a high speed multi-network (ATM, DQDB FDDI). A multi-network is an assemblage of communication equipments and softwares that enable the user to consider the overall network as one resource only. Today's problematic is the administration of these heterogeneous systems (multi-network) after being that of networks architecture and their interconnection. As multi-networks grow in size at a rapid pace, the various components and multi-network users interact in increasingly complex ways. These complex interactions imply intelligent, automated, efficient, and integrated management. Emphasis lies on QoS management in the FDDI management domain.

0-7803-1811-0/94$04.00©1994IEEE

MANAGEMENT STRUCTURING

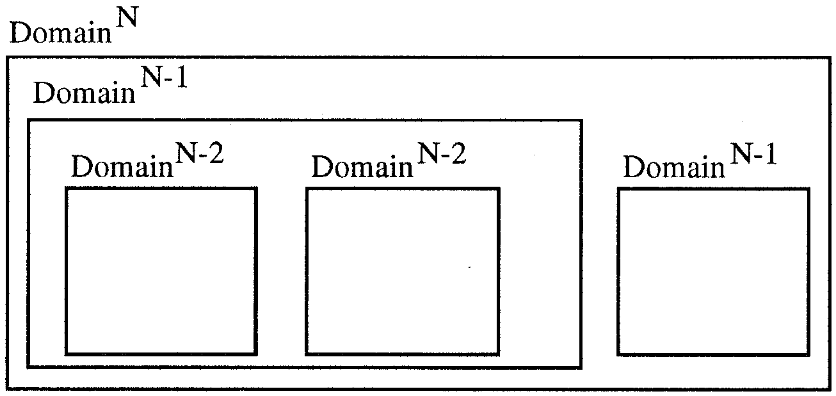


Figure 1: hierarchy of domains; levels N, N-1 & N-2

Management of large scale networked systems is a complex task due to the number and the diversity of resources and activities attached to them. In practice, this complexity may be reduced by structuring the management system into management subsystems and by distributing management responsibilities over these subsystems. This leads to multiple coexisting managements that can be autonomous or interacting in hierarchical or peer-to-peer relationships. Therefore, we have defined a methodology for management organizing and structuring which consist of two structuring principles:

1. Use of *domains* as the unit of management structuring;

2. Make a clear distinction between management policies (i.e. the objectives of management) from the resources and activities being managed (i.e. managed objects).

The domain concept provides a flexible means for grouping resources and defining management boundaries. Furthermore, it allows hierarchical structuring in that domains can be members (subordinates) of other domains (see figure 1). Management tasks are provided by individual domains which cooperate to achieve the overall management objectives.

In addition, the interactions between domains (hierarchical or peer-to-peer) may be formally defined which make easier to set up the corresponding protocols. Another advantage of structuring the real world onto management domains is to easily support dynamic changes without disruption of the entire management system, e.g. new domains joining the system.

The domain concept has been used by a number of groups in the USA for security purposes [2], [3], [4]. They are also used by other research groups (e.g. in the DOMINO Project [5]), [6] and standards [7] for explicit grouping of resources. While in previous works domains were either managers or managed, our domains are composed by a set of managed resources but also encapsulate the components performing management. This choice allows for less flexibility when building the management system but allows the management activity to be applied the same way at all levels of the hierarchy and thus make easier its automation.

Management components and resources may then be related to construct domains according to different criteria that may be relative to the contained managed resources (type, location, functionality, ownership, ...) or to the contained management components and objectives (management function, applied policy, authority, ...) [8].

DOMAIN INTERNAL STRUCTURE

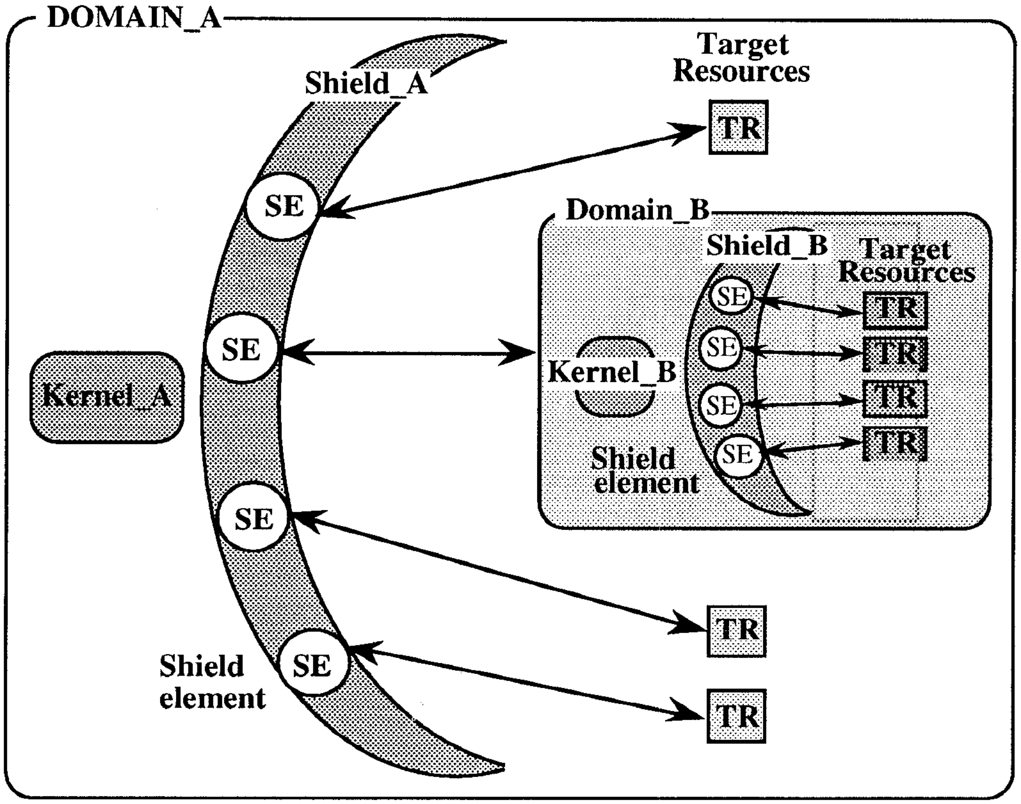


Figure 2. Internal structure of a domain

Our goal is to ensure a uniform management of all resources and activities in a networked system. These last range from hardware resources such as lines and switches to software resources such as databases. Therefore, the second structuring principle has been defined to separate the management activity from the resources that are subject to management. This allows us to define a uniform model for the management activity in order to automate it.

As stated in the previous section our domain consists of a set of resources and a management part applying a certain policy or an aspect of a policy (see figure 2), the management part of a domain is called the *Domain Management System* ("DMS"). We identify two basic parts of a DMS: the abstract representation of the managed resources (called the *Shield*) and the managing part (called the *Kernel*). The kernel performs control actions over the managed resources through their representation in the shield. Managed resources are called *Target Resources*. Each target resource is either a real resource (e.g. a switch, a printer, etc.) or another domain of a lower level in the management hierarchy. The shield is introduced for openness and reusability purposes; it hides the resource's functional interface and reveals the management interface only. It is the domain's component which provides the demarcation between the management activity and the resources being managed.

The shield has no autonomous management activities. It presents a *uniform*, *selective* and *abstract* view of the domain's target resources to the managing kernel. Indeed, the kernel may need to have a uniform view of a number of different resources. The abstract view allows to hide irrelevant details from the kernel and the selective view allows to restrict the kernel access to the part of the resource interface relevant to the management objectives of that kernel.

MANAGEMENT ACTIVITY WITHIN THE KERNEL

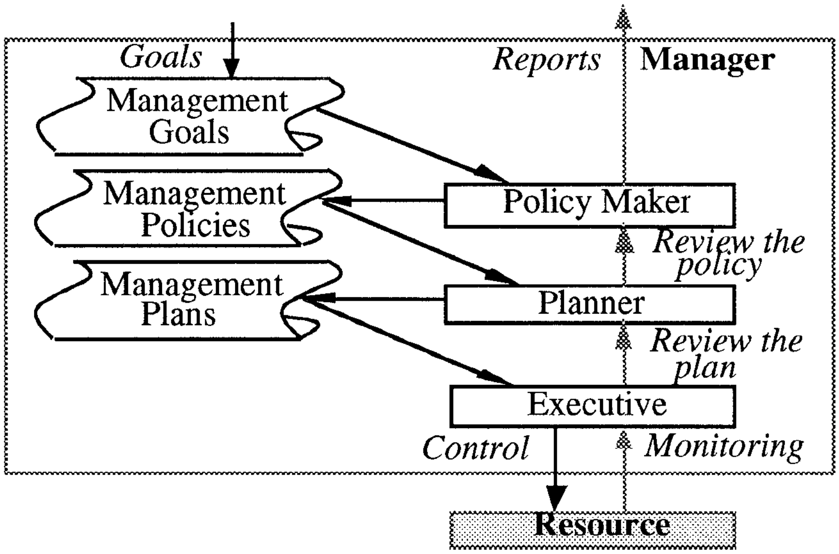


Figure 3 : Kernel Internal Structure

Management is commonly the process by which the manager *monitors* and *controls* resources to meet management objectives [9] (we call *goals*). Monitoring allows one to get knowledge about the resource's behaviour and to check their consistency with respect to the goals. The manager *reports* to its superiors about its management particularly when it fails to achieve the goal. A managed resource can be as simple as a real-resource or as complex as a management system itself (a domain). In the last case, the managed-domain's manager acts autonomously according to its own management policy but also responds to received control commands from higher level managers. Such management role delegation is applied recursively through the different levels of the management hierarchy. In the following, we want to refine the model of management delegation by clarifying the internal structure of the kernel.

The kernel receives management goals from higher level kernels. A management goal is a statement about what is to be achieved. The kernel achieves the management goals by performing control actions over the managed resources according to a management plan. A management plan is a procedure of actions which can be deterministically evaluated at the time it is to be executed. Because making management plans from a set of management goals can be a very complicated task, the concept of management policy is introduced as an intermediate step. Management policies are general statements about how the kernel will achieve the management goals, and are used to ease decision making by restricting the set of solutions to a problem from the range given by the management goals to a more easily handled size. We distinguish two modes of operation within the kernel: - *proactive management* stimulated by arrival of goals from higher level managers and, - *reactive management* on events detection or arrival of notifications from managed resources and/or lower level kernels.

Proactive management consists of three distinct phases. These are policy making, plan making, and plan executing. These management phases are respectively handled by the manager active entities illustrated in figure 3. Giving a management goal, the entity responsible of policy making derives management policies that are used by the planning entity to make plans. The obtained plans may be executed immediately to achieve some of the management goals, or may be stored for execution in response to some situation as part of a continuing goal.

The Executive is responsible for executing management plans. It performs control actions upon the managed resources and, sends management sub-goals to the lower level kernel if the managed resource is a domain.

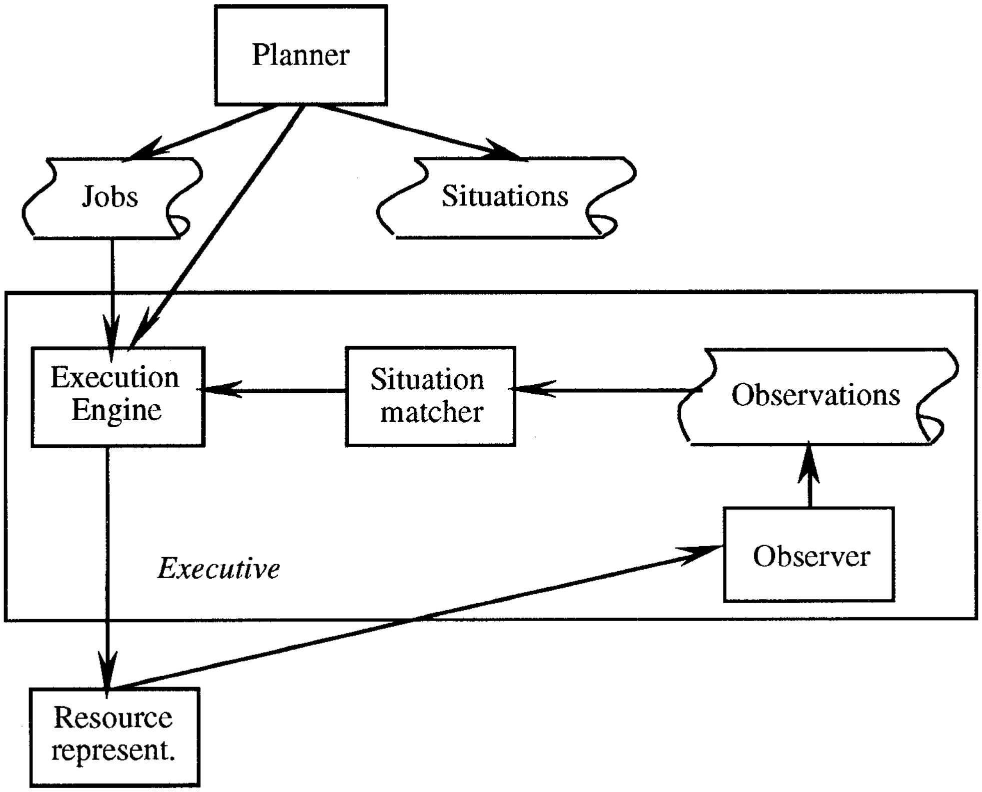


Figure 4: The Executive Internal Structure

Reactive management consists of executing control actions (according to a stored plan) when a particular situation is detected at the managed resource's level. This may give feed back to the higher level management depicted by the dashed arrows, in figure 3. For example, when the executive is unable to execute a job, a message is sent to the Planner asking it to review the plan or to make an alternative one which can be executed. The same process applies to planning and policy making.

The executive is a central component in this management scheme. Its main task is to execute jobs which are one of two parts of the management plans introduced so far. The other part consists of situation specifications. Jobs are executed, by the Excution Engine, as soon as they are created by the Planner, or in response to some situation. The latter case is possible by matching the observed state of the system with stored situation specifications described in terms of patterns of observations. The observations are provided by the Observer. Jobs are represented by algorithmic blocks and their execution results in a sequence of instructions being sent to shield elements that represent resources or managers of resources.

Management functions (including the five ISO defined network management functions) are then handled by the kernel according to the three-phase process described previously. In addition, the kernel is concerned by meta-management such as booting and reconfiguring the management system (e.g. creation of domains, deletion of resources, etc.). A domain's kernel also provides an interface to the external world (i.e. for interactions that are external to the domain boundary). Indeed, domains may share management of one or several resources. This involves interactions between the overlapping domains materialized by their kernels peer-to-peer negotiation for management consistency (e.g. concurrent management synchronization, management optimization and conflicts detection/resolution).

POLICY MAKING : AN INTERMEDIATE APPROACH

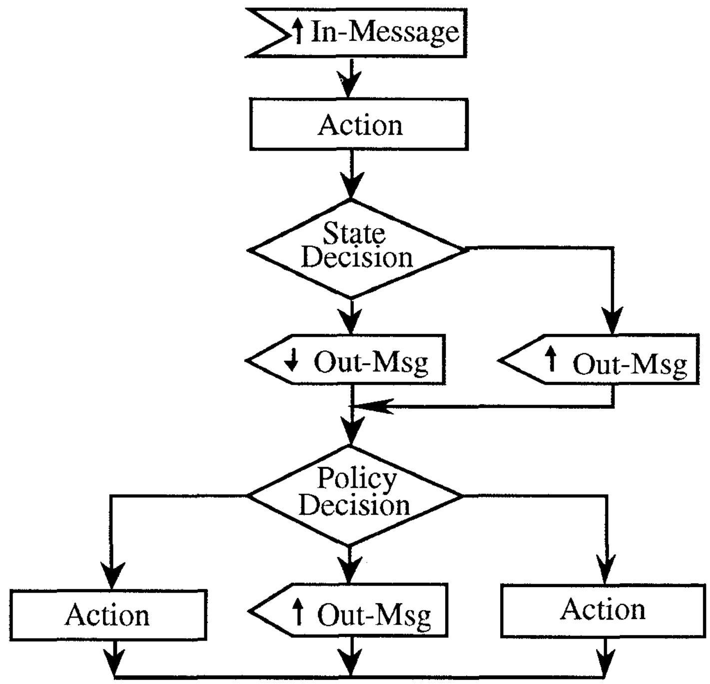


Figure 5 : A plan Template

Our objective is to automate the management process. However, even if the second management phase (i.e. deriving management plans from management policies) can be performed automatically by a machine, the first phase (i.e. making management policy from goals) can only be performed by a human or by a sophisticated program without time constraints. This is due to the complexity of the policy concept (not formally defined). Few works have simplified policies in order to implement them. An example is the Domino project where policies are limited to access rights [10]. Therefore, the full scale model of the goals-policies-plans-actions has been implemented, in the scope of our high speed multi-network management in a limited scenario. We have designed managers in such a way that only a limited and well defined set of goals can be specified, from which a set of policies can be determined automatically and then turned into a plan of actions by applying *Policy Predicates* to a predetermined set of plans in the form of *Plan Template* (see figure 5).

To achieve such an intermediate scheme, we have adopted a bottom up approach. It consists of investigating for each manager which control actions are possible. These actions are then put together to form a sensible and coherent set of sequences of actions (i.e. the plan template). A plan template consists of a tree of action sequences with state and policy decisions at each branch.

Now, from the obtained policy decisions and in conjunction with the set of goals to be given to a manager, we can decide the composition of policies. Indeed, the result of a policy decision can be provided through an evaluation of policy predicates (a deterministic expression of policies). For example, a manager may have to decide between fixing a broken connection or reconfiguring the connection out of use. The factors involved in such a decision may include the cost of the repair in terms of money, time, perceived inconvenience, permanence of solution, etc. The policy predicate will reflect these factors and how much importance the designer places on them.

Policy predicates are represented as explicit calculations to determine a policy decision. This approach has the advantage that purely state-based decisions can be implemented in the same way. In the following, we apply this scheme for goals-policies-plans-actions to the QoS management in the FDDI domain.
The remainder of this paper focuses on the application of our concepts to the management of a high speed multi-network (ATM, DQDB, FDDI).

THE MULTI-NETWORK ENVIRONMENT

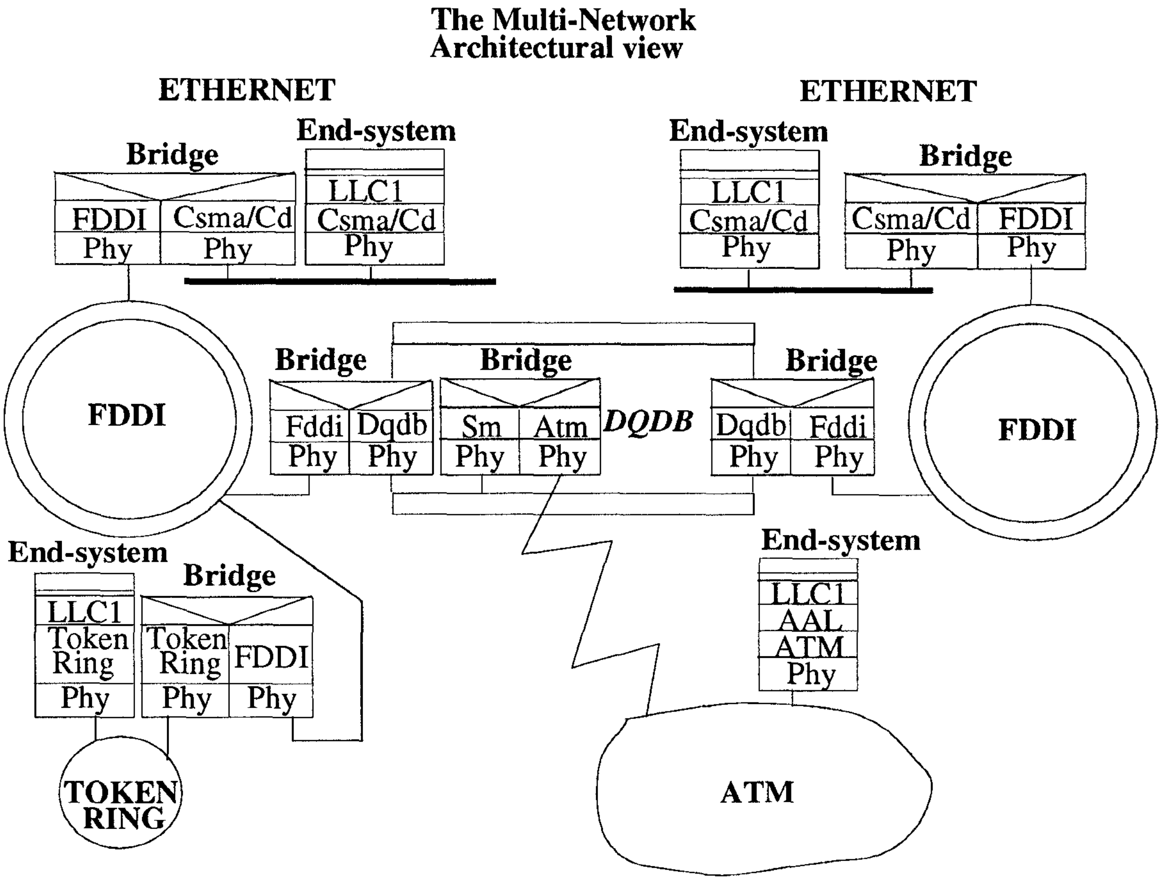


Figure 6: the architecture of the multi-network

While enterprises' telecommunications needs rapidly increase and vary, telecommunications constructors provide only partial and sometimes incompatible solutions to these increasing requirements. Besides, even if the offer of one constructor may meet all requirements, users do not want to depend on one manufacturer only, because of obvious strategic reasons. Moreover, each application requires appropriate network characteristics. Thus, telecommunications structures, already widespread, are based on heterogeneous systems. Such environment is called a multi-network (assemblage of communication equipments and softwares that enable the user to consider the overall network as one resource only).

The development of a multi-network [11][12] (FDDI, DQDB, ATM, Token Ring, Ethernet) demonstrator (figure 6) is carried out at the network department of Télécom Paris. The implementation has been supported by the Eiffel object oriented language [13]. Both Network and MAC interconnection levels have been considered, involving respectively bridges and routers. Here, we assume internetworking at MAC level.

This object-oriented multi-network consists of a DQDB MAN as backbone for FDDI, Ethernet and Token Ring interconnection through bridges or routers, and a BISDN network which has been added lately.

THE MULTI-NETWORK DOMAINS

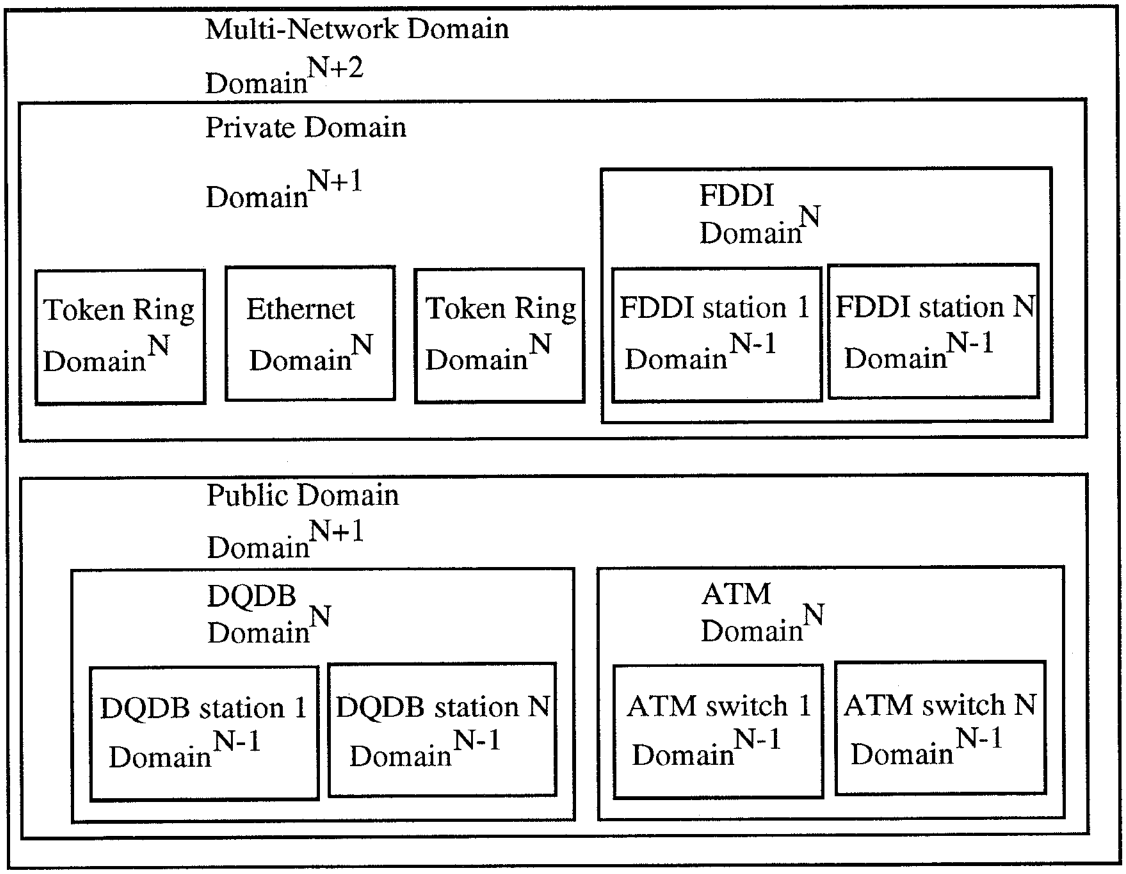


Figure 7: structuring of the management of the multi-network into domains

Which criteria have been applied in order to obtain this structuring of domains within our multi-network?

Location criterion: Within a public or private domain, we apply this rule to separate the different subnetworks in domains (Ethernet, Token Ring and FDDI for a private domain) and (DQDB, ATM for a public domain).

Public and private domains have been separated according to the *policy* criterion. Indeed, these two domains do not share the same responsibilities and requirements. For example, accounting is much more crucial for public than private domains.

Within the FDDI domain, subdomains have been defined according to the *organisational* criterion. Each FDDI station has its own management part named SMT (Station ManagemenT) [14]. The global FDDI management is realised through cooperation between SMTs. This rule is also applied to both the DQDB and ATM subnetworks which management entities are distributed among stations (DQDB) and switches (ATM) respectively.

Let us enhance the FDDI station domain.

FDDI STATION DOMAIN AND ITS SUBDOMAINS

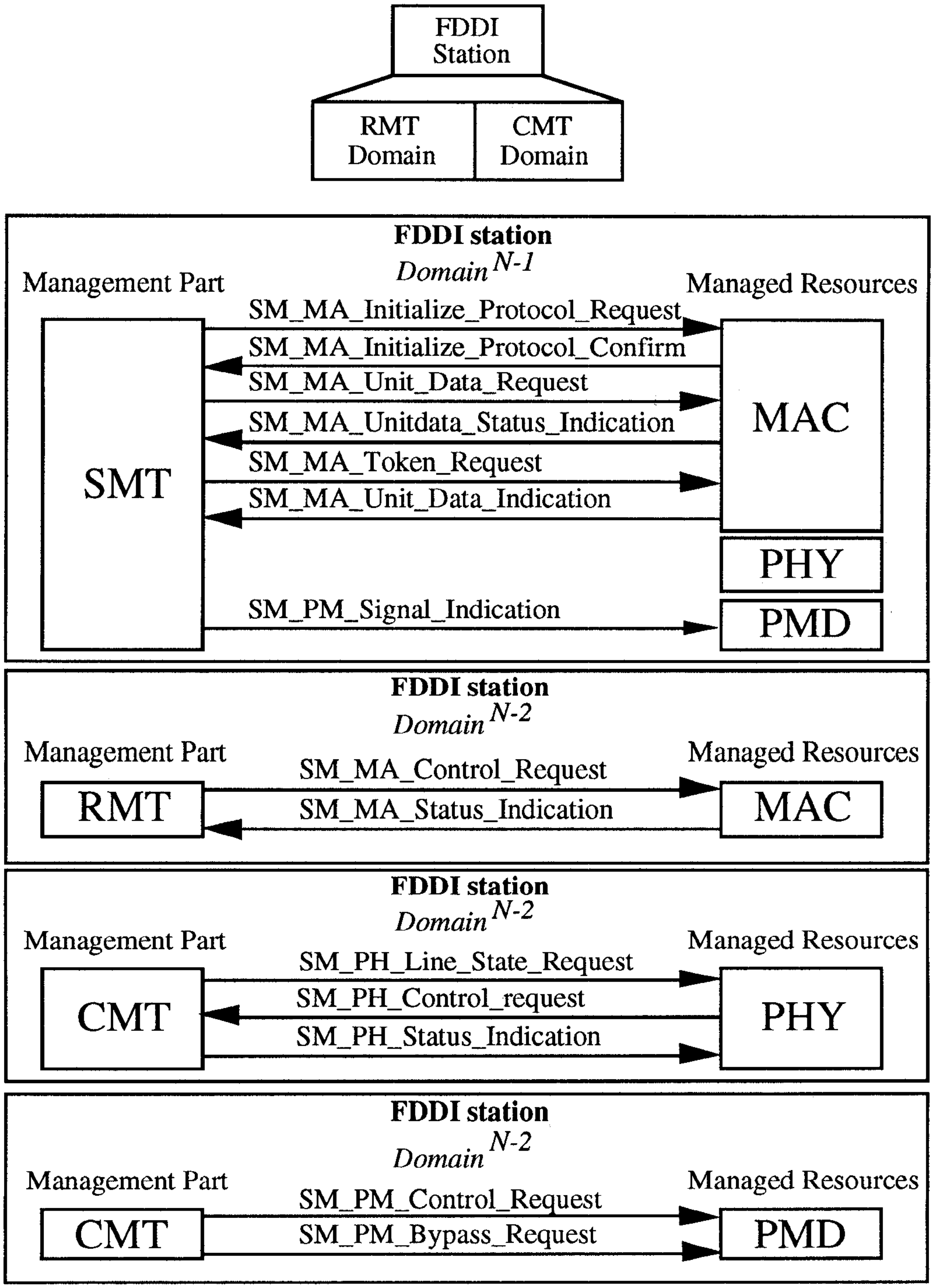


Figure 8: structuring of SMT.

The FDDI station domain is also subdived into an RMT and CMT domains. The applied criterion is the manager functionality criterion. The MAC layer manager (RMT: Ring ManagemenT) is responsible of monitoring MAC operation and takes actions necessary to aid in achieving an operational ring. The physical layer manager (CMT: Connection ManagemenT) controls the establishment of a media attachment to the FDDI network, the connections with other nodes in the ring, and the internal configuration of the various entities within a station.

SERVICES DOMAINS

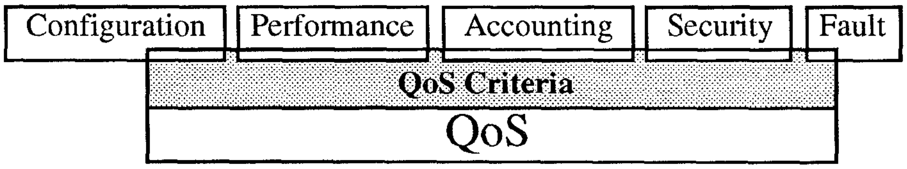


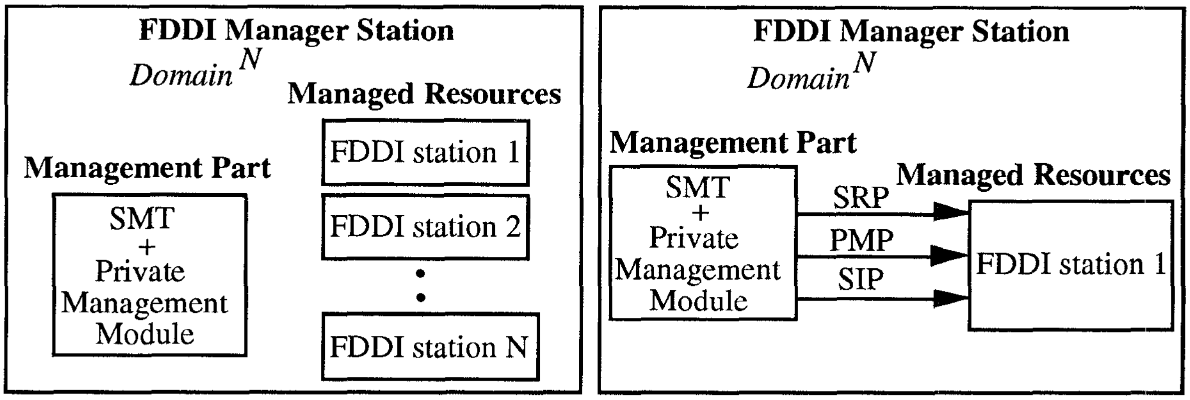
Figure 9

Criterion \ Functional Area	Configuration	Performance	Fault	Accounting	Security	
Availability	X		H,S	X		Service View
Reliability	X	X	H,S	X		
Transit Delay	X	X	H,S	X		
Throughput	X	X				
Transfer Failure	X	X	H,S	X	X	Protocol View
Error Rate	X	X	H,S	X	X	
Non-Sequence /Duplication			S			
PDU Lifetime	X	X		X		PDU View
PDU Size	X	X				
PDU Priority	X	X	H,S		X	
PDU Security	X					

Figure 10: Qos Criteria and OSI Management Functions

After having defined management domains for management of the multi-network architecture, it is required to define management services which also correspond to domains. Configuration, performance, fault, accounting, security are examples. The applied criterion is the manager functionality criterion. In this area we focus on QoS as this function is generic. The entities modeling QoS serve all the management functions as shown in figures 9&10.
Quality of Service has to be considered from the beginning of the *multi-network design* to meet the user's requirements for efficient communications. Computation of the global QoS is obtained according to QoS of each network element constituting the multi-network.
QoS has several views, namely, the end-user's view, the network operator's view, the network designer's view, the network owner's view and the network supplier's view. Each of these views also corresponds to a domain. A methodological approach is necessary in order to satisfy each view' requirements and obtain a convergence of management between these domains [15].
Moreover, the perception of QoS is influenced by the fact that customers demand for higher grade of service which is often of subjective evaluation [16].

We propose to translate and quantify all these elements in terms of computable criteria in order to *measure, control, analyse* and *manage* QoS.
We have identified eleven important QoS criteria which characterize three levels: The service level which focuses on the service, the protocol level which expresses the protocol's treatments and the PDU level translating the Protocol Data Unit' behavior [17].
At the service level, identified criteria are availability (service state), reliability (service rate), transit delay (service time) and throughput (service capacity).
At the protocol level, defined criteria are transfer failure (protocol state), error rate (protocol rate), and non-sequence/duplication (protocol level).
At the PDU level, selected criteria are PDU lifetime, PDU size, PDU priority and PDU confidentiality.
We distinguish hardware faults (H) from software faults (S) as these latter require more sophisticated detection and diagnosis means.



SRP: Status Report Protocol
PMP: Parameter Management Protocol
SIP: Status Information Protocol

Figure 11: QoS of a global FDDI subnetwork

How can we measure QoS of a global FDDI subnetwork, according to QoS of each FDDI station? For that, it is required to apply the following computation rules:

Availability =F1(Availability of each station)= F1(state of each station).

Reliability = F2(Reliability of each station)= F2(F3(error rate, loss rate)).

Transit Delay: In order to measure transit delay, a scenario is required as no information is provided within the FDDI MIB (Management Information Base). The FDDI network manager orders the SMT entity of the station where it is located to generate a frame. At reception of this frame, it calculates the time spent by the frame to go around the ring The occurrence of this operation depends on the polling period.

Throughput= Σ (Throughput of each station). The throughput of a station is the number of frames transmitted by this station.

QoS criteria specified at the service level (availability, reliability, transit delay and throughput) are sufficient to provide the global state of the managed resource.

For computation of the global QoS (QoS of a subnetwork) it is necessary to collect and aggregate QoS of each station on the FDDI subnetwork.
This QoS information is requested through SMT protocols (SRP, PMP, SIP). These protocols enable an SMT entity (The SMT of the FDDI Manager) to request any remote SMT entity (agent for the FDDI manager):

- SRP is performed by any FDDI station to periodically announce its status that is useful in managing an FDDI ring;
- PMP enables access to the FDDI management information base of a remote station. Possible operations are get and set;
- SIP is used to request and provide in response, an FDDI station's configuration and operating information.

At the multi-network level, the hierarchical manager responsible of a private domain requests the FDDI manager in order to know the global FDDI network state. Standardisation of criteria computed by each network component facilitates cooperation and decisions making at any management level.

EXAMPLE OF QoS MANAGEMENT IN THE FDDI DOMAIN

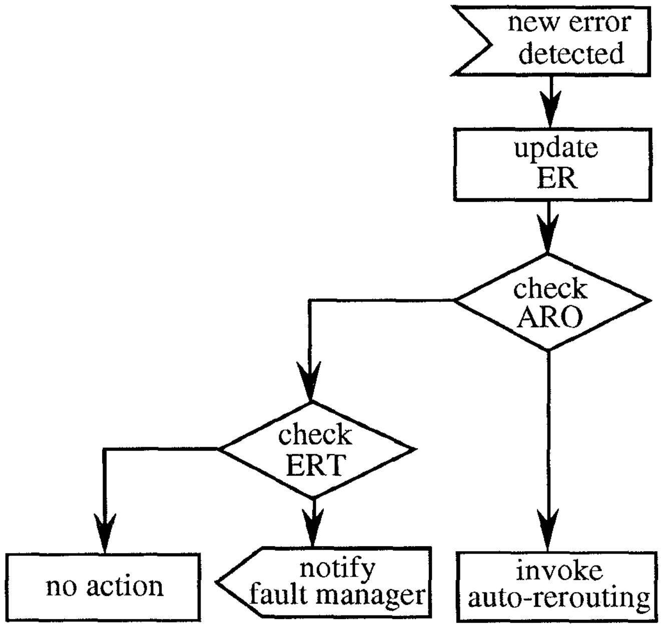


Figure 12

The first step to apply the previous approach to achieve automatic QoS management within the FDDI domain consists of determining the management kernel ingredients that are :

- State attributes, i.e. status of the managed resources in this domain,
- Policy attributes such as thresholds and management options, and,
- Actions.

The second step consists of setting up a plan activation process. A plan is a set of actions to be activated when certain situations occur. A situation is determined by the values of the previous attributes. Typically, a plan activation process is handled as follows:

- Incoming messages (e.g. event notification);
- Update of state attributes;
- Check against policy attributes;
- Invocation of action(s), i.e. the corresponding plan.

QoS Manager State Attributes :

- Error rate (ER);
- Transit delay;
- Availability;
- Throughput.

QoS Manager Policy Attributes :

- Information flows automatic rerouting option (ARO);
- Error rate threshold (ERT);
- Availability threshold (AT).

QoS Manager Actions :

- Automatic rerouting of flows;
- Renegociation of T_req value;
- Notification of Fault manager;
- Deactivation of a station;
- Particular flow reject.

CONCLUSION

Management of large scale networked systems is a complex task due to the number and the diversity of components and activities attached to them. In order to cope with this complexity, we have introduced a methodology for management structuring. The well defined structures can then be used by multiple management applications without need to perform these structuring tasks themselves. Domains have been introduced as a means of delimiting management boundaries. They allow to separate management tasks and to abstract from management details. Management tasks are then provided by individual domains which cooperate to achieve the overall management objectives. The domain concept allows hierarchical structuring in that domains can be members (subordinates) of other domains. Furthermore, management interactions between domains are well defined (cooperative or peer-to-peer) which make easier to set up the corresponding protocols. Another advantage of structuring the real world onto management domains is to easily support dynamic changes without disruption of the entire management system, e.g. new domains joining the system. These concepts have been applied to structure the management of a high speed multi-network (ATM, DQDB, FDDI). The FDDI domain and its subdomains have been enhanced. For that subnetwork, we have focused on a particular management function, the QoS. We have shown how to compute QoS criteria and how to automate this QoS management.

ACKNOWLEDGEMENT

The Authors gratefully acknowledge Philips Gmbh, Siemens AG and FIT e.V.(Germany), Roke Manor Research Ltd (UK), AEA, MARI Computer Systems Ltd. and APM (UK), Telesystemes (France), and the University of Athens (Greece) for their contributions.

REFERENCES

[1] M. Möller & al. "DOMAINS Basic Concepts for Distributed Systems Management". Proc. of the fifth RACE TMN Conf. on 1991.

[2] D. Erstin. "Controls for Interorganization Networks". in Gligor 1987, pp 249-261.

[3] D. M. Nesset. "The inter-Authentication-Domain (IAD) Logon Protocol, (Preliminary Specification and Implementation Guide)". Lawrence Livermore National Laboratory, P.O. Box 808, Livermore, CA 94550, 6 May 1988.

[4] D.A. Gomberg. "A Model of Inter-Administration Network User, Authentication & Access Control". Mitre Corporation, Washinton C3I Division, 7525 Colshire, Drive, McLean VA 22102, MTR-87W00003, Dec 1987.

[5] M. Sloman and J. Mofett. "Domain Management for Distributed Systems". Integrated Network Management, vol. 1, B. Meandzija and J. Westcott (Eds.), North Holland, Amsterdam 1989.

[6] B. Wang, D. Coffield and D. Hutchison. "Database/domain approach to distributed systems management". Computer Comm. 1989.

[7] Information Proc. Syst, OSI Systems Management Overview, ISO/IEC DIS 10040, 1991.

[8] R. Boutaba, A. Benkiran. "A Framework for Distributed Systems Management". Proc. of NETWORKS 92, Int. Conf. on Computer Networks, Architecture and Application, Trivandrum (INDIA), October 1992.

[9] R. Boutaba. "A Methodology for Structuring Management of networked Systems". IFIP Conf. on AIP Techniques for LAN and MAN Management,Versailles (France), April 93.

[10] J.D. Mofett, M.S. Sloman. "Policy hierarchies for distributed Systems Management". Domino internal Report Arch/IC/6. Imperial college of science Technology and Medecine. Dept of Computing, June 92.

[11] G. Calvignac, N. Simoni, S. Znaty. "ATM/DQDB Interconnection at Cell/Slot Level for Connectionless Services". Interworking in Broadband Networks Book, IO Press, June 93.

[12] S. Znaty, N. Simoni. "An Object Oriented Methodology For Realization and Management of a High Speed Gateway". IFIP Conf. on AIP Techniques for LAN and MAN Management,Versailles (France), April 93.

[13] B. Meyer. "Object Oriented software engineering". Prentice Hall, 88.

[14] G. Milligan, F. Ross. "FDDI Station Management (SMT)". Draft proposal, June 92.

[15] W. Tawbi, L. Fédaoui, E. Horlait. "Dynamic QoS Issues in Distributed Multimedia Systems". 2nd Int. Conf. on Broadband Islands, Athens, June 93.

[16] N. Simoni, S. Znaty. "QoS: From Definition to Management". 4th IFIP Conference on High Performance Networking. Liege (Belgium), Dec 92.

[17] N. Simoni, S. Znaty. "Interconnection of High Speed Data Networks: Contribution of Dynamic Control and QoS". Computer Networks and ISDN Systems Journal, Nov 92.