# Designing a distributed management framework– An implementer's perspective

*M.FLAUW - P.JARDIN*
*CEM Technical Office*
*DIGITAL EQUIPMENT CORPORATION*
*SOPHIA ANTIPOLIS - 06901 - FRANCE*
*Tel: +33 92 95 54 26 Fax: +33 92 95 58 48*
*flauw@vbo.mts.dec.com - jardin@vbo.mts.dec.com*

**Abstract**

The distributed organisation and topology of telecommunications networks impose management solutions which are themselves distributed. The direction of such solutions is clearly indicated by the ITU-T TMN architectural framework which is fundamentally based on an Object Oriented paradigm.

The development of distributed solutions poses real technical challenges to vendors. This paper addresses the issues that an implementer of management solutions must consider. It discusses the perceived requirements and trade-offs that have to be faced in the design of a distributed framework.

The essence of DIGITAL's distributed Telecommunications Management Information Platform (TeMIP) is presented.

**Keywords**

Distributed management, Object-oriented framework, TMN, implementation

## 1. INTRODUCTION

The size and complexity of telecommunications networks and their continuing evolution have created interesting challenges for network managers and network management solution developers. The business and political environment creates tremendous pressure on network operators and telecommunications service providers towards delivering maximum quality of service at minimum cost. This has generated requirements for integrated management environments in which the various Operations Systems involved will exchange faster and safer critical, quasi real time, information.

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

In such a context a number of alternatives are offered. Pragmatism forces us to recognise that no single overarching approach can be adopted and even in standards based approaches (e.g. OSF DME), a number of different APIs, modelling languages and messaging systems are proposed. The interworking of these different technologies translates into a number of gateways.

In this complex and evolutionary situation, DIGITAL with its distributed Telecommunications Management Information Platform (TeMIP), has taken both a well architected and pragmatic approach. TeMIP is an evolution of the DECmcc framework that is specifically designed for managing telecommunications networks.

This paper offers an implementer's viewpoint which shows the constraints and often conflicting requirements that a management framework implementer must face.

## 2. AN OBJECT-ORIENTED FRAMEWORK

### 2.1 Object orientation for network management

Significant research and development are being directed to the area of computing enterprise management. Its importance manifests itself in a number of conferences and publications. The pure Object Orientation initially defined and used for programming languages has been (sometimes loosely) adapted for defining management solutions. It has been researched by consortia such as RACE, TINA-C [1], formalised through standardisation activities (ISO [2], ITU-T [3], X/Open [4], ETSI [5], T1 [6]) and realised through implementations such as the DIGITAL TeMIP framework presented in this paper.

The object-oriented analysis methodologies proposed for problem analysis and design [7] have inspired the development of management solutions. In particular, a specific methodology has been defined for these contexts (ITU-T M.3020 [8]). The approach was retained for TMN as it contains essential characteristics such as:

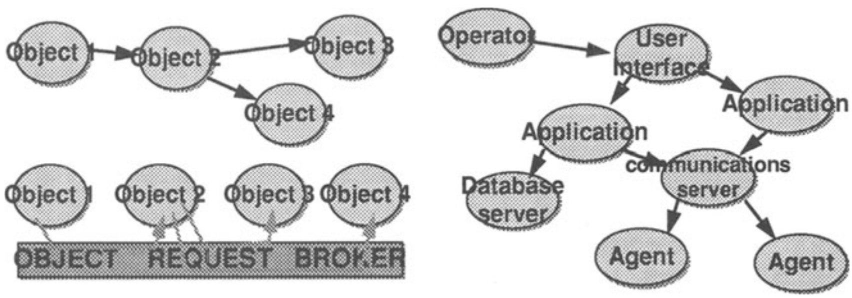
- the ability to define generic specifications that can be adapted to local situations with the concepts of inheritance and polymorphism,
- the ability to hide implementation details by decoupling specification from implementation aspects and focusing on object interfaces (a concept of great interest for the integration of legacy systems).
- the abililty to present different levels of abstraction. This methodology provides a 'zooming effect' allowing to gradually focus on more and more detailed aspects

Defining a management solution in an object-oriented fashion imposes the recognition of the natural dichotomy that prevails in this context:

- The managed resources are generally physically dissociated from the managing systems. The OSI management [2] and the SNMP [9] models have formalised this by introducing the concepts of *Manager* and *Agent*. An object- oriented approach will consist of modelling the managed resources as objects and making them visible via agents.
- On the Manager side, the managing application(s) may themselves be modelled and implemented as objects. They may be distributed as suggested by the ODP approach [10] as a set of interactive objects. These application objects may be very different in nature, e.g. computing components, database servers, user interfaces, communication servers, etc.

Consequently , as depicted in Figure 1-b, a management solution can be designed as the interaction between a number of fairly different objects. TeMIP is a globally object-oriented framework in which all these object classes are modelled under the single modelling approach defined by EMA (DIGITAL's Enterprise Management Architecture [11],[12]). Each class is implemented via one or several '*Management Modules*' (MMs).

The implementation of these classes as generic re-entrant MMs gives a set of building blocks that may gradually be loaded ('*enrolled*') into the framework. An idle system is simply a juxtaposition of classes/ modules that may potentially communicate with each other. An application actually becomes alive at run time when the relevant classes begin to interwork by invoking each others' services via an *Object Request Broker* (ORB) as depicted in Figure 1 -a.




**a/ information exchange via an object broker**

**b/ full object oriented view of a management solution**

**Figure 1**: Co-operating Objects.

A TeMIP based management solution is a collaborative organisation of class instances. In a monolithic implementation each class is instantiated only once, while in a distributed implementation certain classes may be replicated on the various nodes. Implementation details are further discussed in section 5.

## 2.2 Implementing objects as 'management modules'

One of the fundamental principles of the TeMIP architecture is that each Object (implemented by a Management Module) supports three types of interface: a '*service*' interface which groups the directives used to access to the methods of each object, a '*client*' interface which the object may invoke to access the services of other objects and a '*management*' interface which groups the directives used to access to specific methods dedicated to the management of the object itself (i.e. the Management Module. This approach, which is depicted in Figure 2, is under consideration by TINA-C ( [1], [13]).

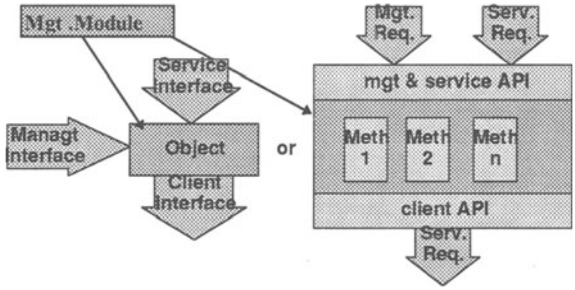



**Figure 2**: Objects Implemented as management modules.

The TeMIP architecture supports a common object specification language to specify the object interfaces and a common Application Programming Interface (API) through which any object gives access to the methods it supports (service or management methods) or accesses the services of any other known objects. This API, as depicted in Figure 3, is actually the Dynamic Invocation Interface of the ORB that finds the location of the invoked service based on the supplied parameters and dispatches the request accordingly. This Object Request Broker is called the *Information Manager* in TeMIP terminology.

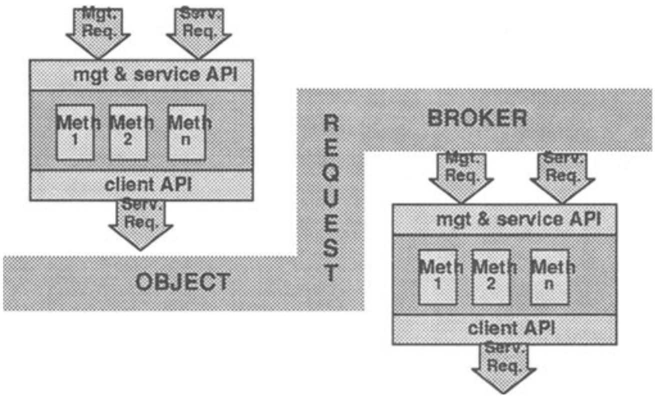



**Figure 3:** Inter Objects communication.

## 3. REQUIREMENTS FOR DISTRIBUTED MANAGEMENT

The development of an object oriented framework is conceptually satisfying but is only useful for the management of telecommunications networks if it fulfils the constraining requirements of such an environment. The essential difficulty in developing integrated management solutions lies in the accumulation of stringent functional and non functional requirements which will influence the design and implementation strategies such as:

- Geographical span: The size and geographical span of these networks impose the development of solutions that allow the functions to be partitioned and located as close as possible to the systems they manage. The management of telecommunications networks is generally partitioned and hierarchical. The management framework must be flexible enough to cope with various topologies.

- Magnitude: The management solution allows a large number of users (several tens) to monitor and control a very large number of resources (hundreds of thousands of object instances).
- Scalability: The network size, configuration and technologies keep changing. The management systems must be able to track and support these evolutions. The introduction of newer technologies must be possible in a stepped approach and without disrupting the service.
- Reliability: The deployment of distributed solutions with some replicated components should provide a form of network fault-tolerance that hardware fault-tolerant systems alone cannot satisfy.
- Openness and ability to integrate legacy systems: The continuing evolution of technologies is accompanied by their long duration, resulting in very heterogeneous environments. Openness can therefore be interpreted as the ability to adapt to an heterogeneous environment with a strong preference for standard-based solutions when applicable.
- Access control/Security: The retained trust model implies that the system is protected at the boundary i.e. all security checks are done at User Interface or Gateway level. The distributed solution must be protected against malicious and erroneous users, while the intra network protection may be reduced.
- Performance: it is expected that a distributed solution will bring obvious benefits in terms of load distribution and throughput. It is realised that, by their nature, distributed topologies may entail slight degradations in terms of response times (due to hopping). This impact must be minimised.
- Manageability: Distributing a system implies adding an additional degree of complexity. This must be taken care of at the management level.

## 4. THE APPROACH TO DISTRIBUTION

Distributing the management solution is essential. A number of techniques may be used each with some advantages and some inconveniences. Ideally, one unified approach should be adopted when the objective is software reusability and performance. In practice, the existence of legacy systems, the multivendor environment and the lack of consensus on any particular solution have led to a number of often overlapping proposals. Since one of the prime requirements is the ability to integrate solution components, this situation has created an unfortunate potential for the proliferation of gateways.

### 4.1 The ideal situation: The universal 'interoperable interface'

Achieving maximum integration and reaching a good level of performance, may be obtained by adopting an universal approach based on a common modelling technique and a minimum set of 'reference points' which translate into well-defined interfaces.

The issue has been identified in the TMN architectural model (M.3010 [3] and the NMF architecture [14]) as one of defining '*interoperable*' interfaces between co-operating components.

The ideal interoperable interface is object oriented. It should be topology independant (WAN or LAN based), compact (support of wildcarded operations), flexible (support solicited/sync. and unsolicited/async messages), efficient (support of atomic requests), secure, etc.

### 4.2 The actual situation: A versatile integration framework

Despite its obvious merits, the use of one single unifying global architecture can no longer be realistically considered in the TMN context. For historical reasons and diversified requirements, the ideal interface was never really agreed at the standard level. Instead, several variants emerged both at modelling level (OSI GDMO [15], SNMP SMI [9] or CORBA IDL [17]) and at stack level (CMIP, SNMP, RPC over OSI or IP). The support of multiple legacy systems additionally imposed a range of proprietary protocols, and thus the logical conclusion was to abandon the idea of a universal interoperable interface.

Some consortia such as the NMF [16] are proposing a series of options that leave the solution designers to make their choice based on environmental constraints and operational objectives. It endorses the OSF DME model ([18],[19]) which decouples the intra/inter application aspects (DME framework) from the manager-agent interface (Network Management Option) based on:

- The CORBA [17] or RPC models [10] which have been designed for handling synchronous type requests. They neither fully support complex interactions e.g. with atomic semantics nor, for the time being, provide satisfactory support for unsolicited information (event notifications).
- The manager-agent models ([2],[9]) which reflect the fact that management operations are fundamentally asymmetrical. This presents some drawbacks when two systems need to interwork as peers [14].

The solution designer will actually tend to organise his solution as the co-operation of '*technology or integration islands*', each of which offers a high level of internal homogeneity and consistency. The technology provider will have to offer a well architected integration framework that allows the interworking of these islands via a series of gateway mechanisms.

### 4.3 Gateways issues

Frameworks must implement multiple gateways and proxy type mechanisms in order to support the various approaches actually used in the marketplace. In some cases, the retained approaches are functionally overlapping and present the unfortunate characteristic of having adopted different modelling languages and underlying protocol stacks.

Integrating the various approaches requires defining non trivial mapping mechanisms such as those defined to integrate CMIP and SNMP ([20], [21]), or CMIP and CORBA [22]. In a similar vein, the integration of legacy systems, most of which are currently controlled and monitored via formatted ASCII message sets, imposes the nontrivial exercise of developing mapping functions such as the TMN 'Q adaptor' ([3], [23]).

These mechanisms typically imply stack interoperation, syntactic and semantic mappings and specification language translations. This proliferation of gateways leads to a more complex support and management of the solution and slower information transfers.

## 5. DISTRIBUTING THE TEMIP FRAMEWORK

In the context where a management solution becomes a patchwork of integrated islands coupled by various gateways, an integration paradigm must be retained for each island. Various design centers may be retained.

The visual integrator approach allows supporting various technologies in parallel but provides minimum to no application interworking capabilities. The tightly coupled mode of integration based on an unifying but constraining achitecture maximises applications synergy and reusability [24]. Both approaches have been retained in TeMIP.

The tightly coupled integration based on the EMA architecture (see Section 2), defines a Management Modules (MMs) hierarchy with *Access Modules* (AMs), which provide the connectivity with the agents/managed resources, *Function Modules* (FMs) which provide value added services and *Presentation Modules* (PMs) which interface with the users (human beings or applications).

Whatever approach is retained, the magnitude of such networks imposes the partitioning and maintenance of different contexts within the integration islands. This section discusses how the EMA based islands can be distributed in response to the magnitude and scalability requirements (see also [25]).

A monolithic TeMIP application is called a *director*. In a distributed topology, each node becomes a director as depicted in Figure 5.

### 5.1 Remote Call request interface

Distributing the framework entails a remote interobject interface ([26],[27]). This interface is referred to as the 'call request' interface. It actually offers the services of an ORB (the Information Manager) as described in section 2. It is a dynamic interface in the sense that the call is formally request- independent (the same call procedure is always invoked) and fully qualified by the call parameters which specify the operation (V/verb), the class instance (E/entity) and the operation parameters (P). The key of identifying a given service is the tuple [V,E,P].

The Information Manager processes the request arguments and acts as a client to establish a RPC binding with the appropriate server. Full location transparency is obtained by identifying the director associated with the target instance (and its supporting MM) via the Distributed Name Services of the framework. This is depicted in Figure 4. The reader who is familiar with the OSF DME architecture will realise that this approach is conceptually the same as that defined for the DME.

The dispatching mechanism is based on dispatch tables that are common in all directors. These dispatch tables are automatically updated in all directors when the solution is augmented with new objects i.e. when MMs are extended to offer new services or when new MMs are '*enrolled*' in the framework. The information in the dispatch tables is used to efficiently compute the management module entry point that provides the requested service.

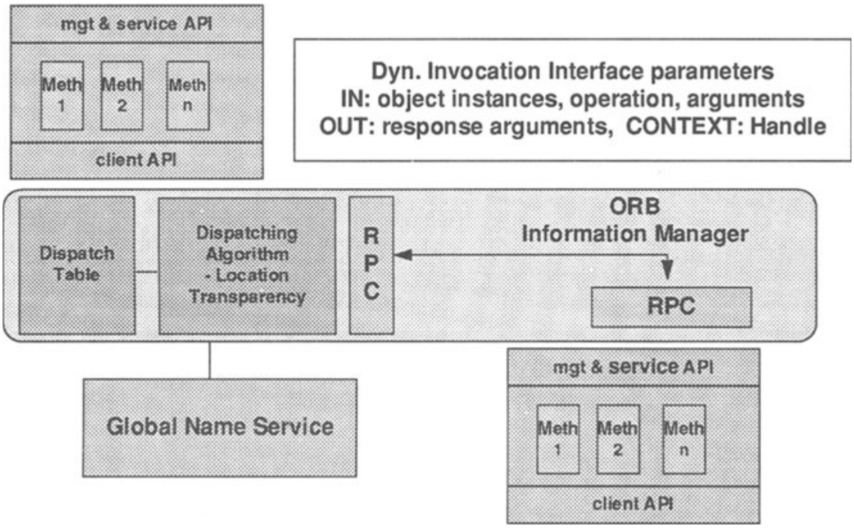



**Figure 4**: Inside the Object Request Broker.

### 5.2 Domain and Entity Access Distribution

In the TeMIP architecture, the 'call request' interface is offered in two variants: a *'call function'* interface used to access value-added services such as those provided by the FMs and a *'call access'* interface used to access the managed object services through the AMs. These two interfaces may both be remotely located for different reasons and will allow the system/solution designer to implement a distributed topology that best serves his operational objectives and environmental constraints.

The use of a remote 'call access' allows the AMs to be located as close as possible to the managed objects/agents. This may be imposed by some technical constraints such as the colocation of the AM software with a non distributed data store or by the type of protocol used between the AM and the Agent. *Entity access distribution* is supported by associating a target object instance with the director that supports its access mechanism in the global name space.

The remote 'call function' is used for load sharing purposes. It allows partitioning work and data, using dynamic grouping criteria realised as *'domains'*. The concept of domain has been designed as a dynamic user-defined grouping of object instances. It can be used to reflect a user's sphere of interest or management policies.

This powerful feature is largely used within TeMIP for historical data collection, alarms monitoring and displaying of information. Basing distribution on domains means that client/server type configurations can be built with optimum and flexible (dynamic) partitioning of the workload. For example, a given FM may be duplicated on different directors and be assigned responsibility for the work that pertains to a certain domain. *Domain based distribution* is supported by associating in the global name space a given director with the domain(s) it is in charge of.

The two forms of distribution are illustrated by Figure 5.

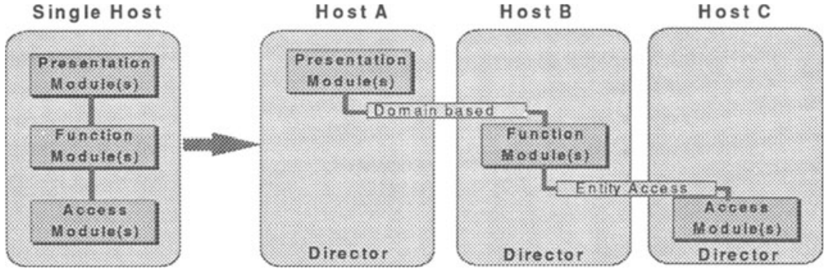



**Figure 5**: Domain based and Entity Access Distribution.

The Object Request Broker determines in real time where is the target module located by identifying the remote director associated with the target object instance or the domain for which the call request is issued.

### 5.3 Data aspects considerations

Two systems can only communicate when they share a common interpretation of the entities they are communicating about. As depicted in Figure 6 this knowledge is generally represented as data which can be subdivided into:

- Metadata representing the classes static information: TeMIP uses a common dictionary model to represent all its metadata. A copy of the dictionary is replicated in each director.
- Configuration data: A distributed system solution must ensure that all managers can reference and access a given object instance. A common network-wide instance name repository is necessary for the ubiquitous and persistent registration of the instantiated classes. TeMIP employs a Distributed Directory Service which provides a global name space.
- Private data: Each object may locally store private data. This data may be made public by the MM via its service interface. TeMIP provides an object-oriented data storage, known as the *Management Information Repository* (MIR). The use of this data storage mechanism is an implementation choice: a designer may decide to maintain the data in non object oriented public files. For example, TeMIP trouble tickets are stored in a relational database and accessible via SQLnet. Note that policy related data (domains, alarm rules, operation contexts etc..) which are modelled as objects fall into this category.
- System configuration data: This data represents the topology of the distributed manager. The MM instance data is maintained as private data by each director and the dispatch tables are replicated in each director.

The automatic replication of global information, the possible replication of functions and the use of fault-tolerant hardware for the support of critical private data are the components of high availability solutions

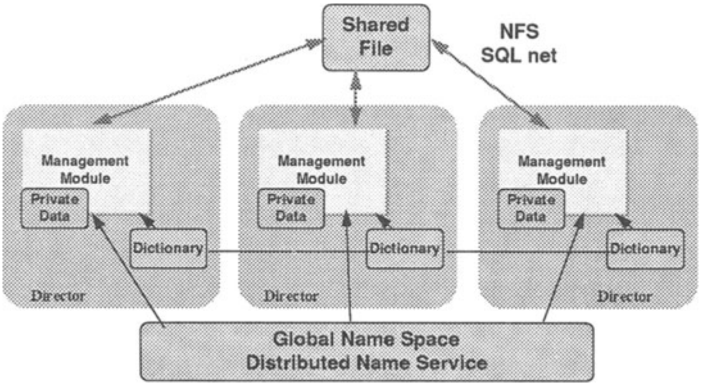



**Figure 6**: Handling data in a distributed topology.

### 5.4 Managing the manager

The management of very large networks imposes complex distributed solutions which themselves need to be managed. A distributed TMN solution, on which the stability of the target telecommunications network relies, becomes itself a network that needs to be configured and managed. This issue is well known and identified in M.3010 as the '*self management*' of the TMN.

A self- management function is quite straight-forward to implement using the TeMIP distributed architecture because of the following essential characteristics:

1. As described in section 2 and Figure 2, the design and implementation of TeMIP as an object-oriented distributed framework implies that each module is itself an object that can be managed via its management interface (see Section 2.2).
2. The concept of domains can be adequately used by the system manager (see section 5.2). The system management activity can be isolated (physically and from a security standpoint) by grouping the managed objects representing the managed directors into a dedicated domain.
3. The distributed TeMIP architecture relies on the services of a particular object, the *Framework MM*, which is in charge of the consistency and stability of each director. This MM is designed to survive system crashes, reactivate long lasting processes and re-establish inter process bindings.
4. The Framework MM also maintains a view of the connectivity with other directors that it interworks with. This is depicted in Figure 7 where the Framework MMs are labelled 'Fn'.

As depicted in Figure 8, the combination of the above features allows the easy management of the TeMIP framework by its own applications. For example, the basic TeMIP Alarm Handling function may be applied to a particular domain composed of the directors and their associated MMs to extract and collect the relevant information from the MMs themselves (considered as managed objects) and build a view of the system behaviour.

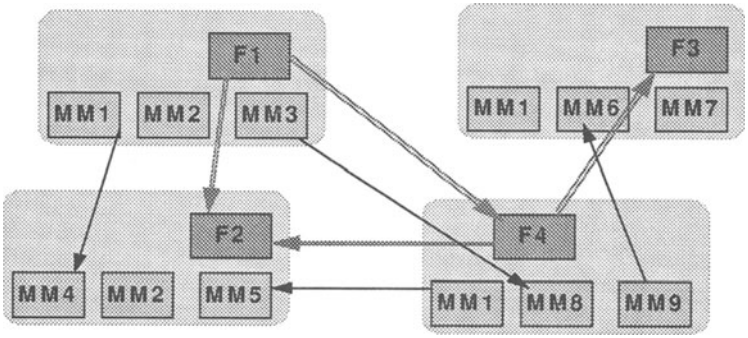



**Figure 7**: Inter Directors communication.

The flexibility of the framework leaves to the system manager the choice of deploying the management application on a separate director or exercising it within an existing director.

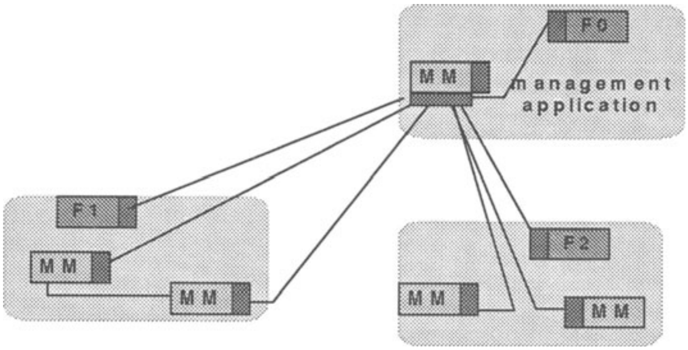



**Figure 8**: Managing the management solution.

### 5.5 Resolving trade-offs

In an ideal world an unified approach should bring significant simplification. In practice the implementation of a distributed framework that fulfils all previously identified requirements can only be complex.

The solution must be available now and nearly as cheap as commodity software. It must support all the latest standards including those still in development and perform well. It must also be user friendly and fully reliable. It must be easy to configure and deploy, scalable and flexible to support network evolution. It should be transparent for application developers, and so on. These are essentially conflicting requirements.

We decided to retain system performance as the prime objective and the driving principle for resolving a number of trade-offs. These include, but are not limited to:

- Grouping activities (domains) to allow load sharing ,
- Supporting a high-speed event communication subsystem,
- Choking off the incoming traffic as early as possible by means of a set of distributed filters,
- Efficient dispatching mechanisms for direct connection between source and target modules. This minimises the round trip delay of the call by avoiding going through intermediate routers or end-point mappers,
- Minimising access time for critical information such as dictionary data or real time data storage,
- Grouping information with concepts such as attribute groups, attribute partitions, event partitions,and support of wild carded operations,
- Strong authorisation and audit trail mechanisms at the periphery (PM level) where time is less critical and reduced internal controls (authorisation and access control on RPC bindings only).

In order to fulfil future requirements when the enabling technologies become available, all essential APIs have been frozen so as to protect the existing applications while allowing rapid swapping of the underlying technology (e.g. dictionary, object data base, communications or name server technology).

## 6. A VERY FLEXIBLE APPROACH - SOME SCENARII

In summary, TeMIP fully exploits the benefits of object-orientation. The use of a common dictionary allows the development of fully data-driven modules (i.e. generic functions which do not need recoding when new classes are added). The use of a distributed name service allows to locate and access to the object instances on or via any director.

Usage and policy independance have been adopted as driving principles for the development of generic modules. No rule or algorithm that normally depend on operational objectives based on local policies are hardcoded which implies immediate code availability and reusability in various environements.

These essential characteristics allow the use of TeMIP in various scenarii:

- Remote User Interfaces (PMs) acting as client running on separate machines can access functions loctaed on a number of 'heavy weighted' servers. This will allow off hour work reorganisation that transfers responsibility to a remote system (critical situation, week ends, etc.). A variant of this scenario can be achieved by means of X-display mechanisms e.g. to support PC-based user interfaces.
- Instrumention of distributed topologies with multiple servers that allow work partitioning can be achieved via domain based distribution. It may be based on:
  - → Policies, operational objectives and skills. A given user has restricted access to the only services that correspond to his skills and job.
  - → Geographical constraints. If the network is split into several regions with a management center for each region, the domains containing the objects related to a given region can be associated with the management center of that region.
  - → Architectural choices such as those retained fro the TMN ([3], [28]).

→ Resource off-loading of functions that are CPU bounds or I/O bounds (database servers) on dedicated systems/directors.

- Developing a front-end approach in which some access modules and communication servers are used to concentrate agent traffic, using entity access distribution to group all entities of a given type on dedicated communications server(s) (OSI/CMIP, SNMP, ASCII etc.).

## 7. CONCLUSIONS

The requirements of large, complex telecommunications networks motivate the research and development of integrated management and distributed solutions. The context is essentially heterogeneous with a slow evolution towards open interfaces.

The design and implementation of distributed frameworks must consequently integrate a number of legacy components as well as emerging de jure and de facto standards which, in many cases, are incompatible.

It is most probable that the ultimate implementation of a fully integrated TMN will actually be a patchwork of internally consistent technology islands interconnected via multiple gateways. The idealistic goal of global integration based on an overarching model will probably never be reached.

Today a few technologies are capable of fulfilling the long list of stringent and sometimes conflicting requirements. DIGITAL's TeMIP is one of these. It was designed from the beginning as an integration framework: It is architected to support multiple protocols and its distributed implementation has been designed to take into account additional essential functional and non functional requirements such as manageability, security and performance.

## 8. REFERENCES

[1] TINA-C, 'Definition of a service Architecture - draft document', October 1993.

[2] ISO 10040/ITU-T X.701: Information Technology - Open Systems Interconnection - 'Systems Management Overview '- 1992.

[3] ITU-T Recommendation M.3010: 'Principles for a Telecommunications Management Network (TMN)' - 1992.

[4] X/Open, 'Systems Management - Managed Object Guide (XOMG)' - X/Open ref. G302 - 1993.

[5] ETSI GSM Recommendation 12.00, 'Objectives and structure of the PLMN management' - 1993.

[6] ANSI T1.210, 'Operations, Administration, Maintenance and Provisioning- Principles of Functions, Architectures and protocols for TMN Interfaces', 1992.

[7] G.Booch ,'Object oriented design with applications', Benjamin Cummings 1993.

[8] ITU-T Recommendation M.3020: 'TMN Methodology', 1992.

[9] Marshall T.Rose, 'The simple Book - An introduction to Management of TCP/IP based internets', Prentice Hall.

[10] ISO JTC1/SC21 DIS 10746-1.2.3, ITU-T Draft Recommendation X.901/2/3: 'Basic Reference Model for Open Distributed Processing - Parts 1 to 3', 1994.

[11] Digital Equipment Corp., 'Enterprise Management Architecture - General Description', Order No EK-DEMAR-GD-001, 1989.

[12] C.Strutt and M.Sylor, 'DEC's Enterprise Management Architecture', Network and distributed systems Management by M.Sloman', Addison Wesley 1994.

[13] P.Jardin, 'The TINA service component approach', TINA-C newsletter, May 1994.

[14] NMF, Forum 004 - 'Forum Architecture' - 1990.

[15] ISO 10165-4/ ITU-T X.722, OSI Management Info. Services - SMI part 4: Guidelines for the Definition of Managed Objects.

[16] NMF, Forum 026 - 'Omnipoint Integration Architecture', Issue 1, July 1994.

[17] OMG, 'The Common Object Request Broker: Architecture and specification' - OMG document No. 91.12.1, Revision V1.1, Dec. 1991.

[18] OSF, 'OSF Distributed Management Environment (DME) Architecture', Open Software Foundation, May 1992.

[19] M.Autrata and C.Strutt, 'DME Framework and Design', Network and distributed systems Management by M.Sloman', Addison Wesley 1994.

[20] P.Kalyanasundaram and A.S.Sethi, 'An application Gateway Design for OSI-Internet Management', Proceedings of the 3rd IEEE/IFIP Integrated Network Management Symposium, 1993.

[21] NMF, Forum TR107 - 'ISO/CCITT and Internet management: coexistence and interworking strategies', 1992.

[22] X/Open, 'GDMO to OMG-IDL translation algorithm', review draft, 1994.

[23] L.Aubertin and T.Bonnefoy, 'Q-adaptor function for customer administration in a switch', IEEE Network Operations and Management Symposium Proceedings February 1994.

[24] P.Jardin, 'Benefits of applying the TMN methodology to Management platforms development', DIGITAL internal paper pending publication.

[25] C.Strutt, 'Dealing with scale In an Enterprise Management Director', Proceedings of the 2nd IEEE/IFIP Integrated Network Management Symposium, 1991.

[26] C.Strutt, 'Distribution in an Enterprise Management Director', Proceedings of 3rd IEEE/IFIP Integrated Network Management symposium, 1993.

[27] Digital Equipment Corp., 'TeMIP Framework System Reference Manual' - order No AA-PD5LE-TE and AA-Q9HGA-E, November 1994.

[28] S.Aidarous, Carey Anderso et all, 'The role of the EML in Network Management', IEEE Network Operations and Management Symposium Proceedings February 1994.

## 9. THE AUTHORS

- Marc FLAUW is a member of the TeMIP technical office. He has driven a number of network management projects. He is one of the key architects of the TeMIP platform.

- Pierre JARDIN is a member of the TeMIP technical office. As one of the architects of the TeMIP platform, he is in charge of AD activities and participates to a number of standardisation bodies such as ITU-T SG4.3, ETSI NA4 and ETSI SMG6.